AD-A065 129 BELL HELICOPTER TEXTRON FORT WORTH TEX F/G 1/3
OPERATIONAL LOADS SURVEY - DATA MANAGEMENT SYSTEM. VOLUME I. US--ETC(U)
JAN 79 R B PHILBRICK, A L EUBANKS DAAJ02-77-C-0053
UNCLASSIFIED BHT-299-099-871 USARTL-TR-78-52A NL

1 OF 2
AD
A065128

USARTL-TR-78-52A

LEVEL II

12

OPERATIONAL LOADS SURVEY - DATA MANAGEMENT SYSTEM

Volume I - User's Manual

Richard B. Philbrick
Alfred L. Eubanks

Bell Helicopter Textron
P.O. Box 482
Fort Worth, Texas 76101

January 1979

Final Report

Approved for public release;
distribution unlimited.

Prepared for

APPLIED TECHNOLOGY LABORATORY
U. S. ARMY RESEARCH AND TECHNOLOGY LABORATORIES (AVRADCOM)
Fort Eustis, Va. 23604

APPLIED TECHNOLOGY LABORATORY POSITION STATEMENT

This report has been reviewed by the Applied Technology Laboratory, US Army Research and Technology Laboratories (AVRADCOM), and is considered to be technically sound.

This program was initiated to design and implement a computer software system for data management of the Operational Loads Survey test data base, allowing the user to select the data to be retrieved, to specify certain processes by which the data may be reduced and/or analyzed, and to choose the mode by which the data will be presented. User options include interactive or batch processing with output options of printing and graphic displays using Tektronix and Calcomp devices.

This program was conducted under the technical management of D. J. Merkley of the Aeronautical Technology Division.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

19 18 6 9 10 15 11 12 16 17 14

1. REPORT NUMBER: USARTL-TR-78-52A	2. GOVT ACCESSION NO.
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): OPERATIONAL LOADS SURVEY - DATA MANAGEMENT SYSTEM. Volume I. User's Manual.	5. TYPE OF REPORT & PERIOD COVERED: Technical Report
	6. PERFORMING ORG. REPORT NUMBER: 299-099-871
7. AUTHOR(s): Richard B. Philbrick, Alfred L. Eubanks	8. CONTRACT OR GRANT NUMBER(s): DAAJ02-77-C-0053
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Bell Helicopter Textron, P. O. Box 482, Fort Worth, Texas 76101	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62209A 1L262209AH76 00 211 EK
11. CONTROLLING OFFICE NAME AND ADDRESS: Applied Technology Laboratory, US Army Research and Technology Laboratories, Fort Eustis, Virginia 23604	12. REPORT DATE: January 1979
	13. NUMBER OF PAGES: 164
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

1L262209AH76 00 166p. nH

16. DISTRIBUTION STATEMENT (of this Report)

BHT-299-099-871

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Volume I of a two-volume report.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Helicopters, Data Bases, Data Reduction, Data Management, Computer Graphics, Interactive Graphics, Signal Processing, Mathematical Analysis.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Operational Loads Survey/Data Management System (OLS/DMS) was designed and programmed as a computer software tool for data management and processing of the Operational Loads Survey (OLS) test data base. With limited modification, the OLS/DMS will accommodate other large, time based, test data bases. The system transfers selected test data to a large, direct access disc file and maintains the data on a semi-permanent

over

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

JOB

20. Abstract (Continued)

basis. Data are retrieved from this file, processed, and displayed interactively or in batch. Plot output is generated on a Tektronix 4014 or an incremental plotter (e.g., Calcomp).

A small sample of available processing options includes amplitude spectrum, harmonic analysis, digital filtering, blade static pressure coefficient, and blade normal force coefficient. This program will accommodate data from multiple sensors simultaneously for processing of functions with two geometric independent variables (e.g., chord and radius). Output options include printout, single or multiple curve X-Y plots, contour plots, and pictorial representation of 3-dimensional surfaces. User input is free field with errors diagnosed where possible. Prompting for available command input is optional.

This report is in two volumes. Volume I is a user's manual and Volume II is a systems manual for assistance in program maintenance, modification, and/or installation.

PREFACE

The AH-1G Helicopter Aerodynamic and Structural Loads Survey conducted under Contract DAAJ02-73-C-0105 was awarded in June 1974 by the Applied Technology Laboratory, U.S. Army Research and Technology Laboratories (AVRADCOM) to produce a comprehensive base of helicopter test data. In particular, measurements were taken of parameters such as airfoil surface pressure, leading-edge stagnation point, local flow magnitude and direction, blade accelerations, bending moments, and the attendant responses in the control system and airframe. The output of 367 transducers was recorded continuously and simultaneously. Over 72,000 separate functions of time were digitized, recorded on digital tapes, and delivered to the Applied Technology Laboratory. The results of the above-mentioned contracted effort are documented in Report USAAMRDL-TR-76-39[1].

The Operational Load Survey Data Management System was developed under Contract DAAJ02-77-C-0053 awarded in September 1977 by the Applied Technology Laboratory (ATL). The software developed under this contract is primarily designed to process data taken during the AH-1G Helicopter Aerodynamics and Structural Loads Survey and other similar test programs. Documentation prepared under this contract consists of two volumes. Volume I provides user instructions and information on the analytical methods used in the software. Volume II, Systems Manual, details the various programming considerations.

Technical program direction was provided by Mr. D. Merkley of ATL. Principal Bell Helicopter Textron personnel associated with the current contract were Messrs. R. B. Philbrick, A. L. Eubanks, and W. R. Dodds.

---

[1]Gerald A. Shockey, Joe W. Williamson, and Charles R. Cox, AH-1G HELICOPTER AERODYNAMIC AND STRUCTURAL LOADS SURVEY, Bell Helicopter Co., USAAMRDL Technical Report 76-39, Eustis Directorate, U.S. Army Air Mobility Research and Development Laboratory, Fort Eustis, Va., February 1977, AD A036910.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Concluded)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Concluded)

LIST OF TABLES

1. INTRODUCTION

1.1 PROBLEM DISCUSSIONS

The principal requirement of Contract DAAJ02-77-C-0053, "Operational Loads Survey Data Management," was the development of a computer software system capable of processing in depth the extensive data base produced by the AH-1G Helicopter Aerodynamics and Structural Loads Survey. However, the software is required to be general in nature and capable of processing other large data bases. The system is required to provide for access, data reduction, and a variety of formats of presentation of helicopter test data. In addition, operation in batch, interactive, and interactive graphics modes is required. Design requirements include execution on an IBM 360 Model 65 with Time Sharing Option (TSO), Tektronix 4014 graphic terminal, Houston Instruments DP-1 incremental plotter, and IBM 2741 typewriter terminal. The mandatory computing language was FORTRAN IV.

Design specifications also included a need for a high degree of user interaction by means of computer generated step-by-step explanations of user inputs, available options, and menus of data available for processing. Most analytical methods commonly used for helicopter test data analysis were to be included and modular design requirements to permit other methods to be added as the need arises. Analytical methods must be selectable by the user in <u>any appropriate combination</u> - a very severe requirement.

1.2 GENERAL SYSTEM CAPABILITIES

The Operational Loads Survey Data Management System (OLS/DMS) meets the above-mentioned design requirements. The system will transfer user-selected data from an initial storage medium, such as digital tape in the case of OLS, to a direct access disc file called the Master File. A directory in the Master File allows automatic, rapid retrieval of the data by the processing/display part of the system which can run in interactive or batch modes. Measured data storage and retrieval features include the following:

- **Measured data are stored on direct access file with a directory for rapid retrieval.**
- **The sampling rate for measured data can be reduced in the storage process.**
- **Measured data can be filtered in the storage process.**
- **There is no program limit to the number of data streams or the length of data streams which can be stored.**
- **Multiple users can access the data simultaneously.**

Various analyses may be performed on the basic or measured parameters contained in the data base and, in addition, several parameters may be derived from basic parameters. The computational capabilities available to the user in the processing program are detailed in Table 1. These analyses and derivations can be performed in multiple dimensions (e.g., time, chord, and radius). Sequences of analyses and/or derivations can be performed on a set of data. For example, Main Rotor Shaft Horsepower could be derived from measured parameters and then the result filtered.

Measured parameters and process outputs can be presented in various ways as shown in Table 2. Simple X-Y plots or multiple-curve X-Y plots are available. Three-dimensional outputs in the form of contour plots and surface perspective drawings in rectangular and cylindrical coordinate systems are available. These output options are available on a Tektronix 4014 terminal in the Interactive Graphics mode of operation or on a CALCOMP or Houston Instrument DP-1 incremental plotter in the batch mode. Printed listings of output are available in either mode.

User command input for both the transfer of data to the Master File and the processing of data from the Master File is free field. For transferring data to the Master File, the input is assumed to be batch mode; however, an interactive utility program exists for creating this input. For the processing of data from the Master File, the input can be interactive or batch. Features which assist the user in generation of command input for processing include:

- The input format is free field.
- Many types of errors are detected in the input process.
- The program can list the available options and/or meaning for each input entry.
- The assistance described in the item above is optional.
- There are defaults for some entries.
- Sequences of commands can be built, stored and executed by name.

Correlation of the data with a rotor azimuth pulse train is available both as part of processing and output scale generation. Most of the derivations and some of the analysis procedures require this azimuth pulse train to be present. However, other processing capabilities, data management features, and the graphics functions can readily be used for nonhelicopter applications.

As an example of application of this system, the user could load the Master File with absolute pressure measurements along

Table 1. ANALYSIS AND DERIVED PARAMETER OPTIONS

ANALYSIS FUNCTIONS	DERIVED PARAMETER FUNCTIONS	
AMPLITUDE SPECTRA	TRUE AIRSPEED	BLADE STATIC PRESSURE COEFFICIENT
HARMONIC ANALYSIS	ROTOR AZIMUTH	BLADE NORMAL FORCE COEFFICIENT
DIGITAL FILTERING	ROTOR RPM	BLADE CHORDWISE FORCE COEFFICIENT
MOVING BLOCK DAMPING	SHAFT HORSEPOWER	BLADE PITCHING MOMENT COEFFICIENT
CYCLE AVERAGING	SHAFT THRUST COEFFICIENT	BLADE DISPLACEMENT
MIN/MAX ANALYSIS	SHAFT TORQUE COEFFICIENT	BLADE LOCAL FLOW MAGNITUDE
	DENSITY ALTITUDE	BLADE LOCAL FLOW DIRECTION

Table 2. OUTPUT CAPABILITIES

PRINTOUT

PLOTS (TO TEKTRONIX 4014 OR CALCOMP):

X-Y PLOTS	MULTIPLE CURVES ON ONE PLOT
	LINEAR OR LOG SCALES IN EITHER DIRECTION
	SMALL DATA 'WINDOWS' CAN BE SELECTED
	TEKTRONIX CROSS-HAIR CURSOR CAN BE ACCESSED TO EVALUATE POINTS
CONTOUR PLOTS	RECTANGULAR OR CYLINDRICAL FORMAT
	CONTOUR INTERVAL AND INITIAL CONTOUR LEVEL SELECTABLE
SURFACE PLOTS	RECTANGULAR OR CYLINDRICAL FORMAT
	SURFACE MAY BE OBSERVED FROM ANY ANGLE

with the measured parameters: Main Rotor Azimuth, True Indicated Airspeed, Boom System Static Pressure, and Outside Air Temperature. The Processing Program could then be used in the interactive graphics mode to derive the Static Pressure Coefficient, C_p, at each Absolute Pressure Sensor location. The user might then request multiple curve plots of C_p with each curve representing a chord position (Figure 1). These plots would be generated on the Tektronix terminal. All plots presented as figures in this manual are calcomp plots and have been reduced to 75 percent size to conform to page size requirements.

Should the data appear to be noisy or contain elements which change from rotor cycle to rotor cycle, the cycle-averaging capability of the Processing Program may be used to eliminate the problem. That is, one representative cycle of the output of a particular sensor is synthesized by averaging several contiguous cycles. After obtaining an averaged rotor cycle of data, the user may choose to recompute C_p and display the results in a contour plot using blade upper surface data at a fixed azimuth angle (Figure 2). The user may next choose to use the program to derive the Normal Force Coefficient, C_n, from C_p for each radial position and display the results in a contour plot or surface plot as shown in Figures 3 and 4 respectively.

Should the user desire an incremental plotter version of the output, a 'SAVE' command feature may be used to store each command on a disc while executing in the interactive mode. Those commands may then be retrieved and executed at some later time in the batch mode so that incremental plotter output may be generated.

1.3 SYSTEM STRUCTURE

The OLS/DMS consists of two major programs, the File Creation Program and the Processing Program, as well as several utility programs. The File Creation Program reads data from some storage medium (digital tape for OLS), selectively transfers data to the Master File, and creates a directory of the data stored on the Master File (Figure 5). The Processing Program retrieves data from the Master File, accepts user commands interactively or in batch mode, processes data, and outputs data in graphic or printed format (Figure 6). Thus, the primary product of the File Creation Program, the Master File, is the measured data input source for the Processing Program.

Four other utility programs are available in the system: the Master File Initialization Program, the Master File Utility Program, the Question and Answer Program to create user input

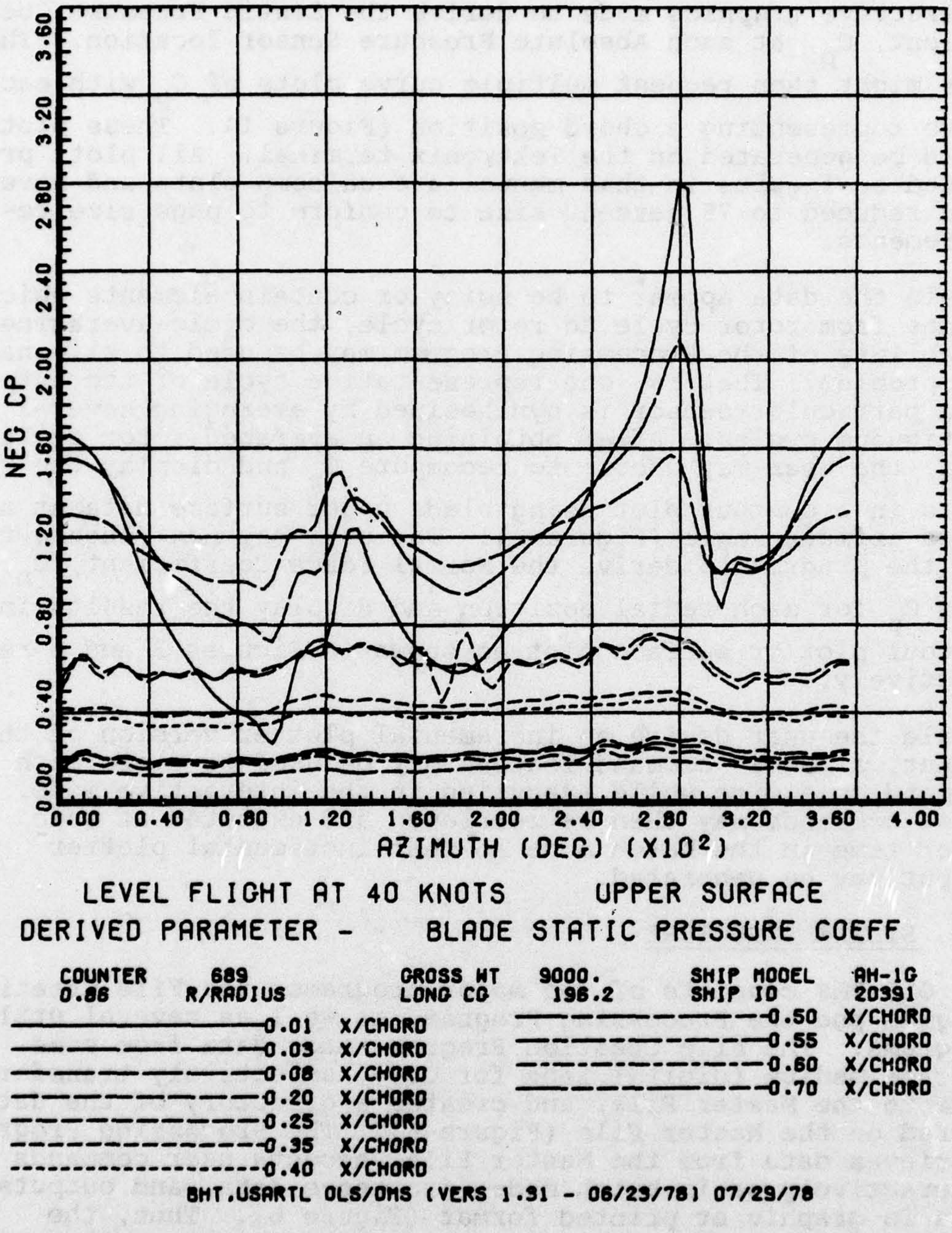


Figure 1. C_p **without cycle averaging for one radial position.**

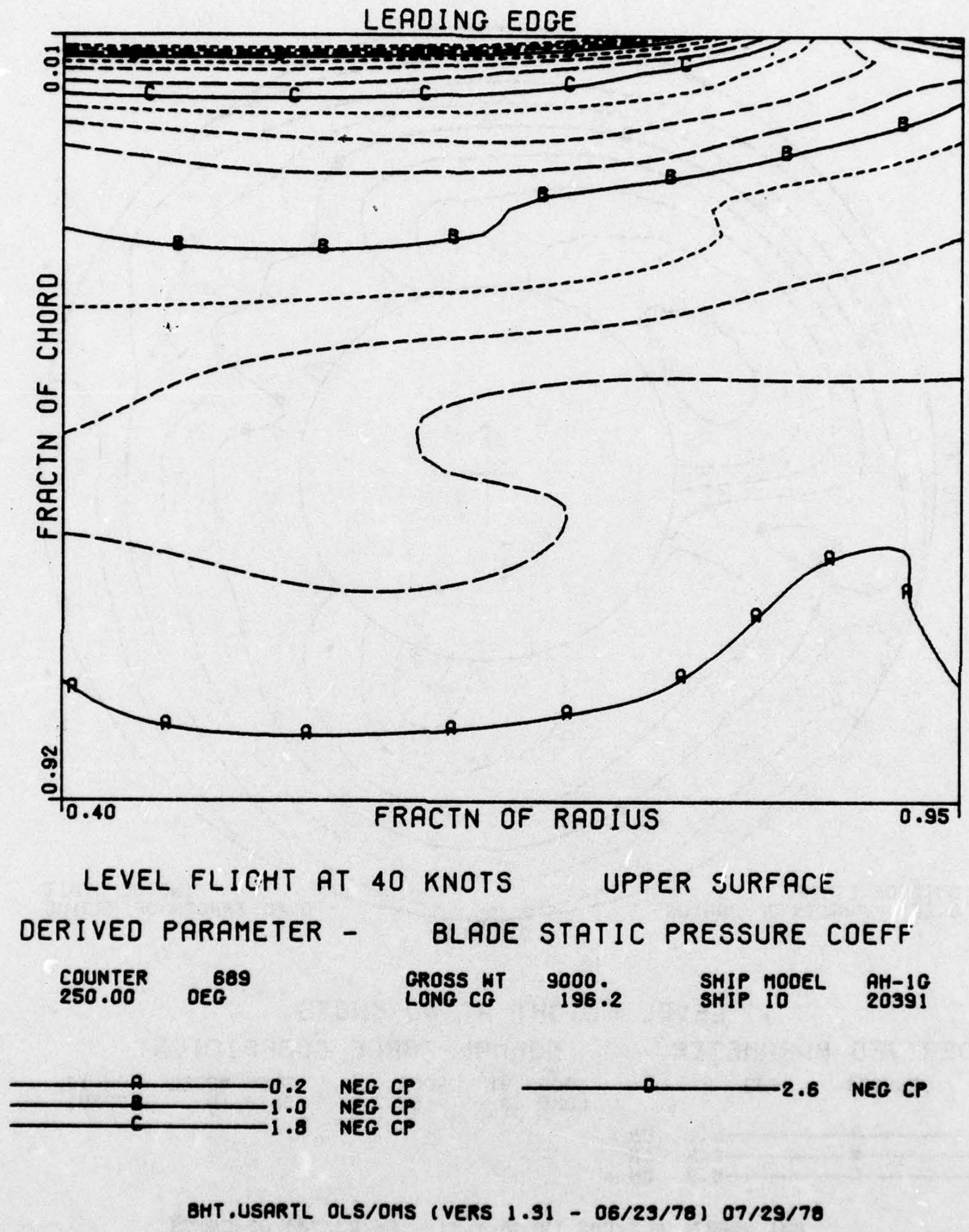


Figure 2. C_p contour for one azimuth position.

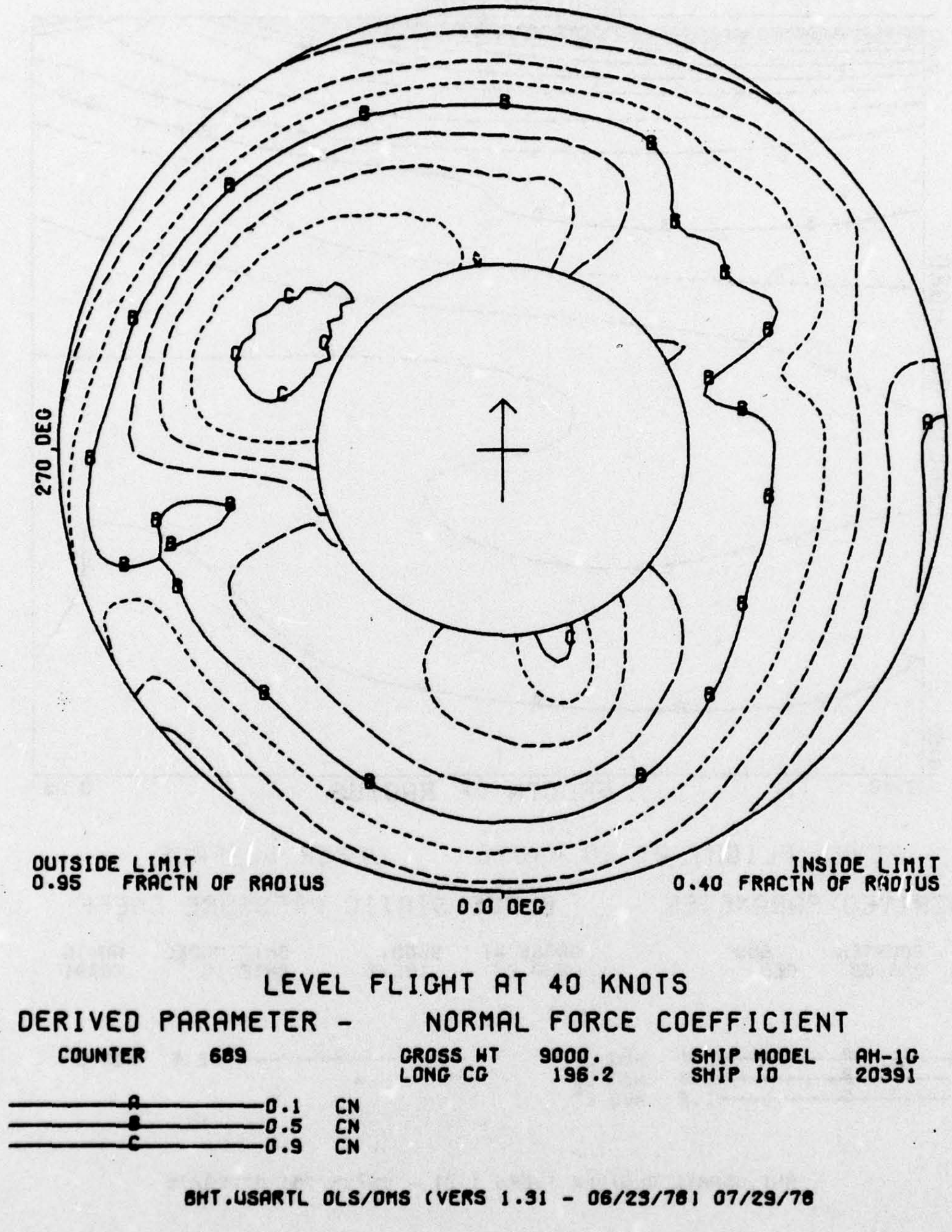
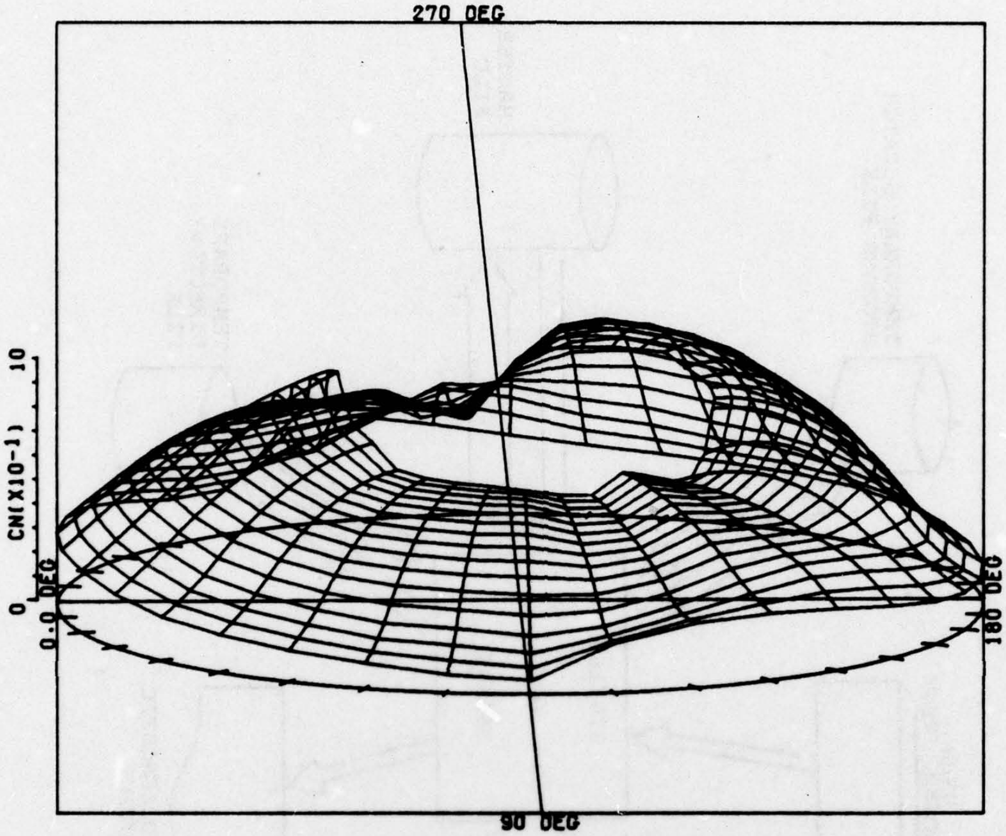


Figure 3. C_n contour plot with cylindrical coordinates.



LEVEL FLIGHT AT 40 KNOTS

DERIVED PARAMETER - NORMAL FORCE COEFFICIENT

COUNTER 689 GROSS WT 9000. SHIP MODEL AH-1G
LONG CG 196.2 SHIP ID 20391

ANGULAR INCREMENT 10 DEG
RADIAL QUANTITY FRACTN OF RADIUS
MAX RADIUS 0.955
RADIAL INCREMENT 0.0370

BHT.USARTL OLS/DMS (VERS 1.31 - 06/23/78) 07/29/78

Figure 4. C_n surface plot with cylindrical coordinates.

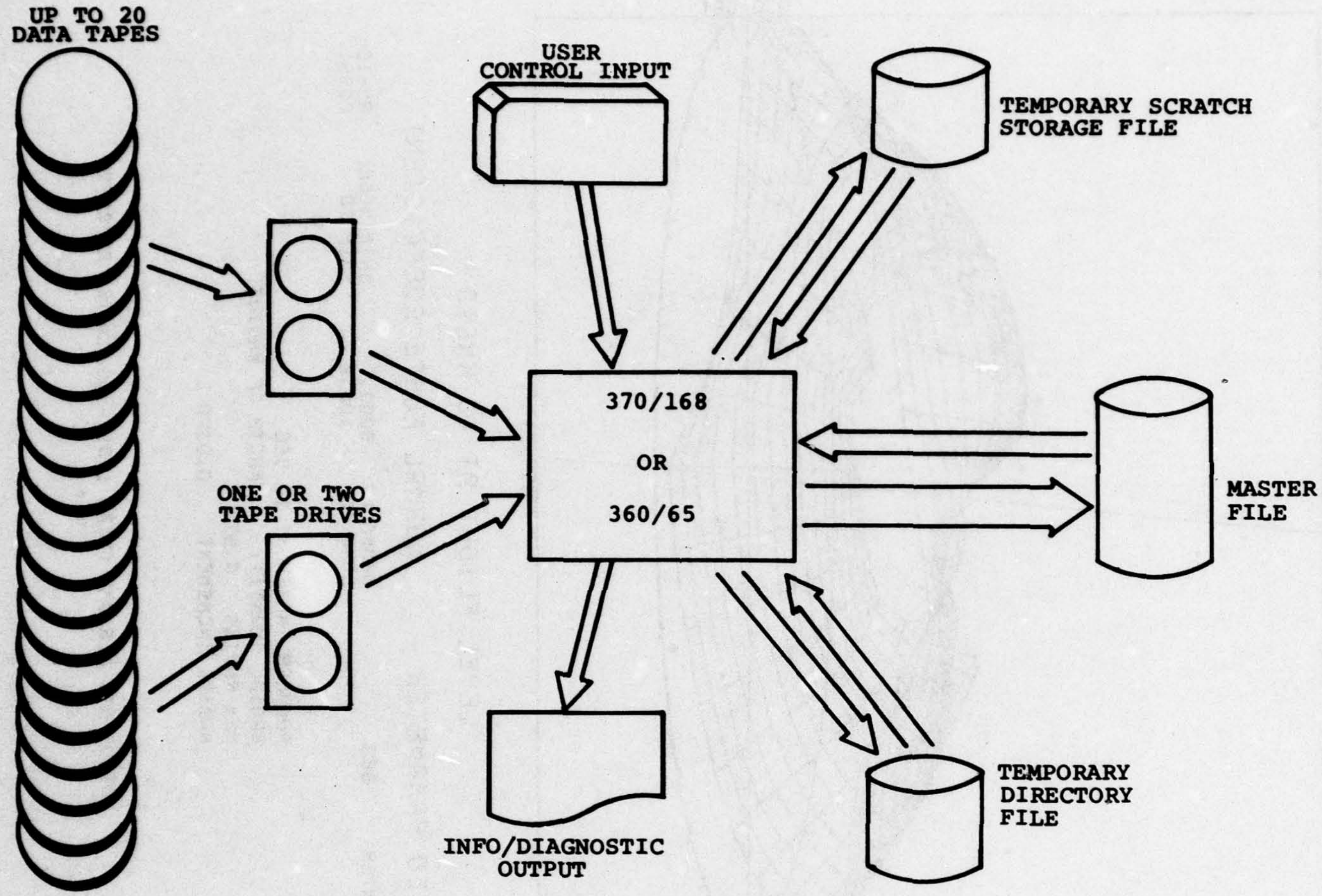


Figure 5. Information flow for file creation program.

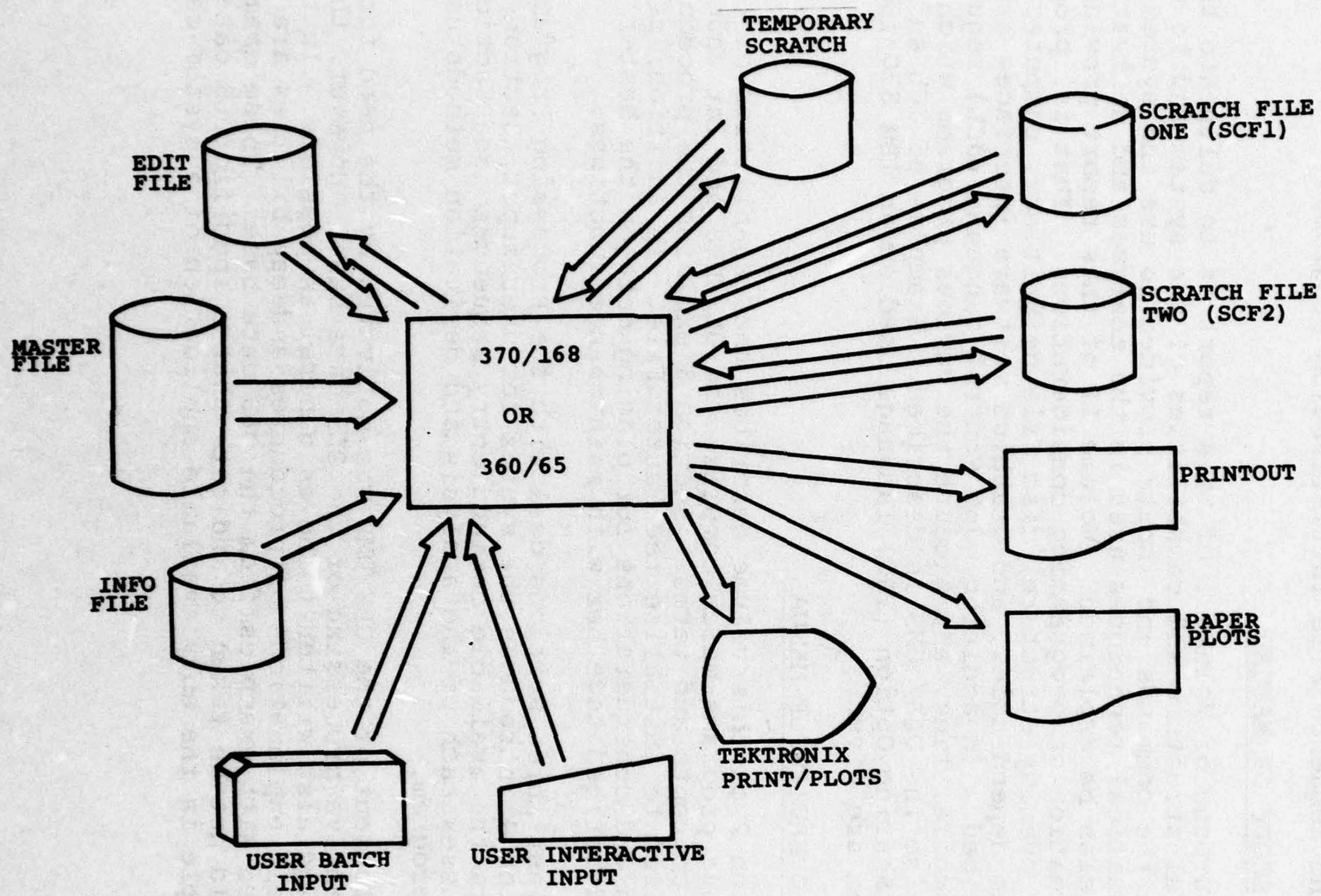


Figure 6. Information flow for processing program.

in a directed mode for the File Creation Program, and the Command Sequence File Initialization Program.

1.4 SCOPE OF MANUAL

The purpose of Volume I of this report is to delineate the general structure and capabilities of the system and to explain the commands and inputs provided to use the system. Mathematical techniques used in the analyses and derivations will also be explained. Volume II of this report provides information on programming considerations. That is, program flow, overlay structure, disc file definitions, computer system dependencies, and graphics software interfaces are described. In addition, Job Control Language (JCL) required to compile, link, and execute the various programs which comprise the OLS/ DMS is described. Examples of JCL and IBM Time Sharing Option (TSO) language used on an IBM 370/168 MVS system are given.

1.5 OVERVIEW OF MANUAL

Section 2 of this volume describes the capabilities of the various programs which comprise the OLS/DMS and introduces many concepts and terms. Section 3 explains the procedures required to initialize the Master File. In addition, procedures and considerations for placing data on the Master File are described together with maintenance functions.

The last three sections deal with the Processing Program. Section 4 introduces the syntax for user input, Section 5 covers the available command entry sequences, and Section 6 discusses each of the analysis and derivation methods used in the program.

Development of the OLS/DMS was motivated by the need for comprehensive processing of the OLS data base. However, the system was also written to be as general as possible. In this manual, explanations of procedures and capabilities are illustrated with examples from the OLS data base. These examples should not be taken to indicate that a specific OLS data base example is the only available application of a system capability.

2. SYSTEM PROCESSING CAPABILITIES

2.1 DATA STORAGE

Data are stored on the Master File in direct access format. The Master File is divided into partitions, each having its own directory. A partition should contain all the data a user may want to access within a run of the Processing Program. A partition is created when data are first written onto the file. A partition may be deleted, additional data can be written onto an existing partition, or a partition can be replaced by a new partition.

Conceptually, the system was designed to provide a separate partition for each user, although several users can use a single partition. When viewed over an extended period of time, multiple, variable-size partitions provide access for several users to a relatively large amount of disc space without requiring the assignment of an exceedingly large amount of total disc space. Each user may add data to the Master File as required and, similarly, each user may delete data from the file as the need arises. However, the data contained in each partition are protected by a password that is assigned by the user when the partition is defined. Each installation should have one person assigned to monitor the Master File and to assure that unnecessary data are deleted. The Master File Utility Program, as described in Paragraph 3.4, is available to support this function.

Data are stored on the Master File as a sequence of discrete values that are indexed by time. These data sequences are often referred to as time histories. Each time history is referenced by item code and counter. The item code is a string of 4 characters that uniquely identifies the output from a particular sensor. A counter is a number between 1 and 32,767 that uniquely identifies a single maneuver. More generally, a counter references a period of time when useful data were taken. An item code/ counter pair may appear only once inside any partition, although it may appear in several different partitions.

There is no specific limit to the number of item codes or counters which can appear in a partition. Similarly, there is no specific limit to the number of data samples which comprise a time history. However, the Master File data set and the partition must be sufficiently large to hold the data.

OLS data is recorded on digital tape in an uncalibrated, integer form and each integer value may be represented using 16 bits. Calibration constants are available on tape to provide conversion to engineering units. However, calibrated data values require 32 bits for computer representation. Data can be stored on the Master File in either the integer or calibrated form. Integer format data require as little as half the space required to store calibrated data. In order to retain as much accuracy as possible, the data is always stored in a calibrated form whenever digital filters are applied before storage.

Each time history stored on the Master File is preceded by an information record which contains useful information about the data stream. Information stored in the information record includes calibration factors, description of the data, units, sampling rate, amount of data present, and other parameters.

2.2 FILTERING AND DATA RATE REDUCTION

The sample rate, or the number of equally spaced data points per second, for data stored on the original medium (e.g., digital tape) may exceed the rate required to depict the highest frequency of interest. For example, the sample rate for the data might be 2048, while the user is only interested in frequency components of 100 Hz or less. If it is assumed that about five data points per cycle are adequate to describe the 100-Hz frequency component, then 512 would be an adequate sample rate.

The File Creation Program provides a capability to reduce a sample rate which a user feels is excessive. The rate is reduced by storing only every Nth sample from the original media, where N is the rate reduction factor or skip rate.

The user may need to use the rate reduction capability to make the sampling rate identical for all item codes of the same kind stored in a single partition. For example, certain absolute pressure sensor data may be recorded on digital tape using 2048 samples/sec, while other absolute pressure sensor data are recorded at 4096 samples/sec. By applying a skip rate of two to the data which had been sampled at 4096, all of the absolute pressure sensor data stored on the partition would be sampled at the same rate.

The highest frequency which can be represented by digitized data is called the Nyquist frequency. This frequency is one-half the sample rate. In reducing the sample rate, the user must consider the possibility of frequency aliasing. Suppose in the first example above that there was a significant noise component at 450 Hz. When the sample rate is reduced from

2048 to 512 the 450-Hz noise component would be aliased to appear at 62 Hz. The File Creation Program provides the capability to low-pass filter the data while transferring it to the Master File. In the pass-band region, the filter used is very flat in magnitude and does not distort the phase (see Figures 7 and 8). See Paragraph 3.2.3 for considerations in using this filter and reducing the sample rate.

2.3 DATA RETREIVAL

At the beginning of execution of the Processing Program, the user specifies the partition of the Master File to be used. Data are retrieved from the Master File as needed by reference to item code, counter, offset from the beginning of data present, and amount of data wanted. This information is specified by the user either directly or implicitly. Data stored in integer format are calibrated and when the data are retrieved, an information record for the time history is also retrieved from the Master File for the program to use in generating various plot or printout labels. The information record alone may be retrieved to generate a display for the user of information about item codes present for a given counter in the partition.

Any number of Processing Program users may access data from one or more partitions of the Master File at the same time. However, if one job is accessing the file with the File Creation Program, no other job or users may access the Master File at that time.

2.4 ANALYSIS AND DERIVATION

The following paragraphs describe the outputs of each of the analysis and parameter derivation capabilities of the Processing Program, while Section 6 describes the method of computation of each of the functions.

The Amplitude Spectrum option uses a Fast Fourier Transform (FFT) algorithm to represent functions of time as functions of frequency. Displays or plots show amplitude as a function of frequency (Figure 9). Input functions of time consisting of well separated relatively pure frequency components yield amplitude sprectra that have been optimized to accurately predict amplitude at the various frequency components. Rectangular, Cosine Taper, r Hanning Window functions can be specified for use in the Amplitude Spectrum option. The beginning user is encouraged to specify the Cosine Taper (COS) window. Windows are discussed in more detail in Paragraph 6.1.

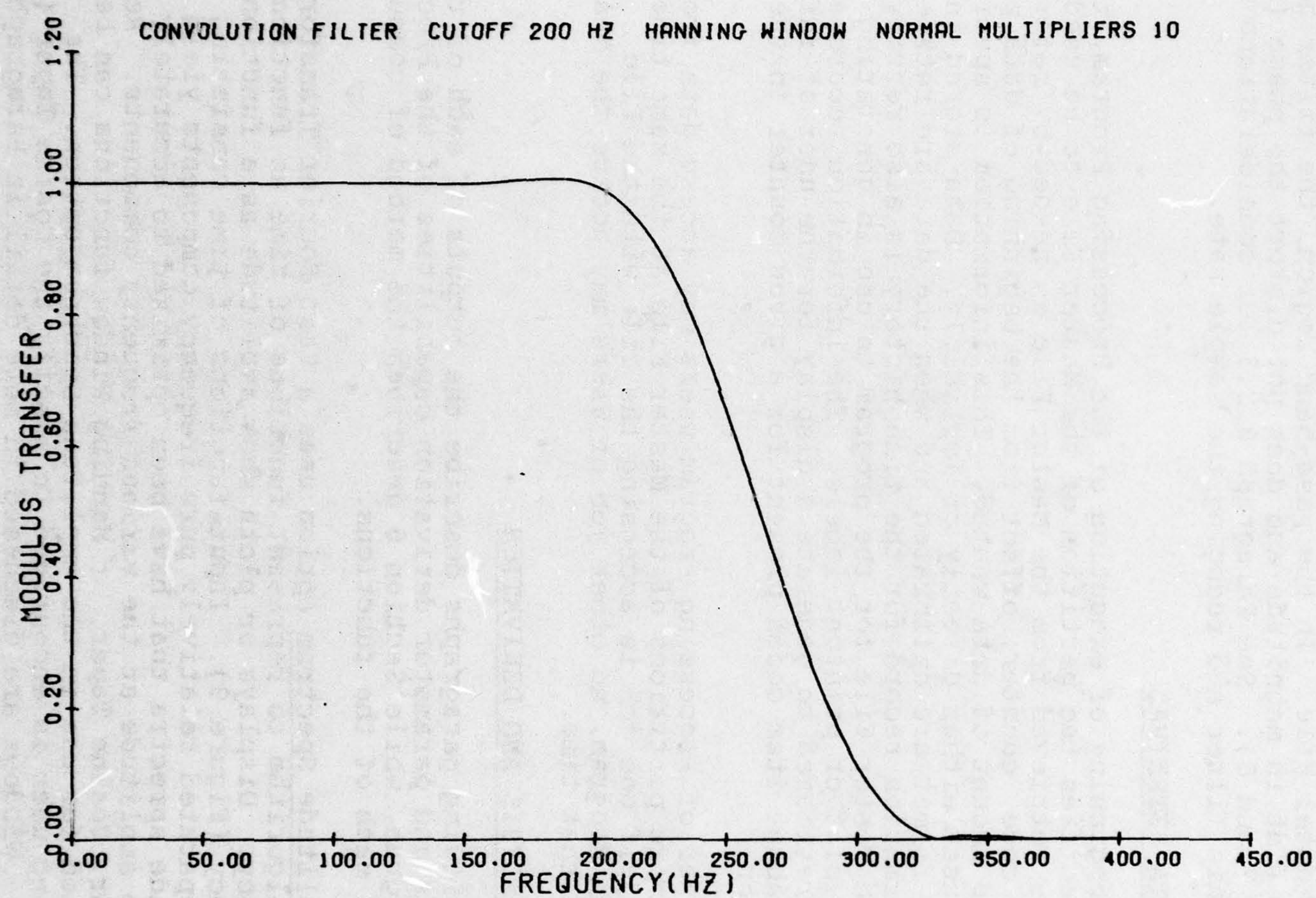


Figure 7. Convolution filter modulus transfer function.

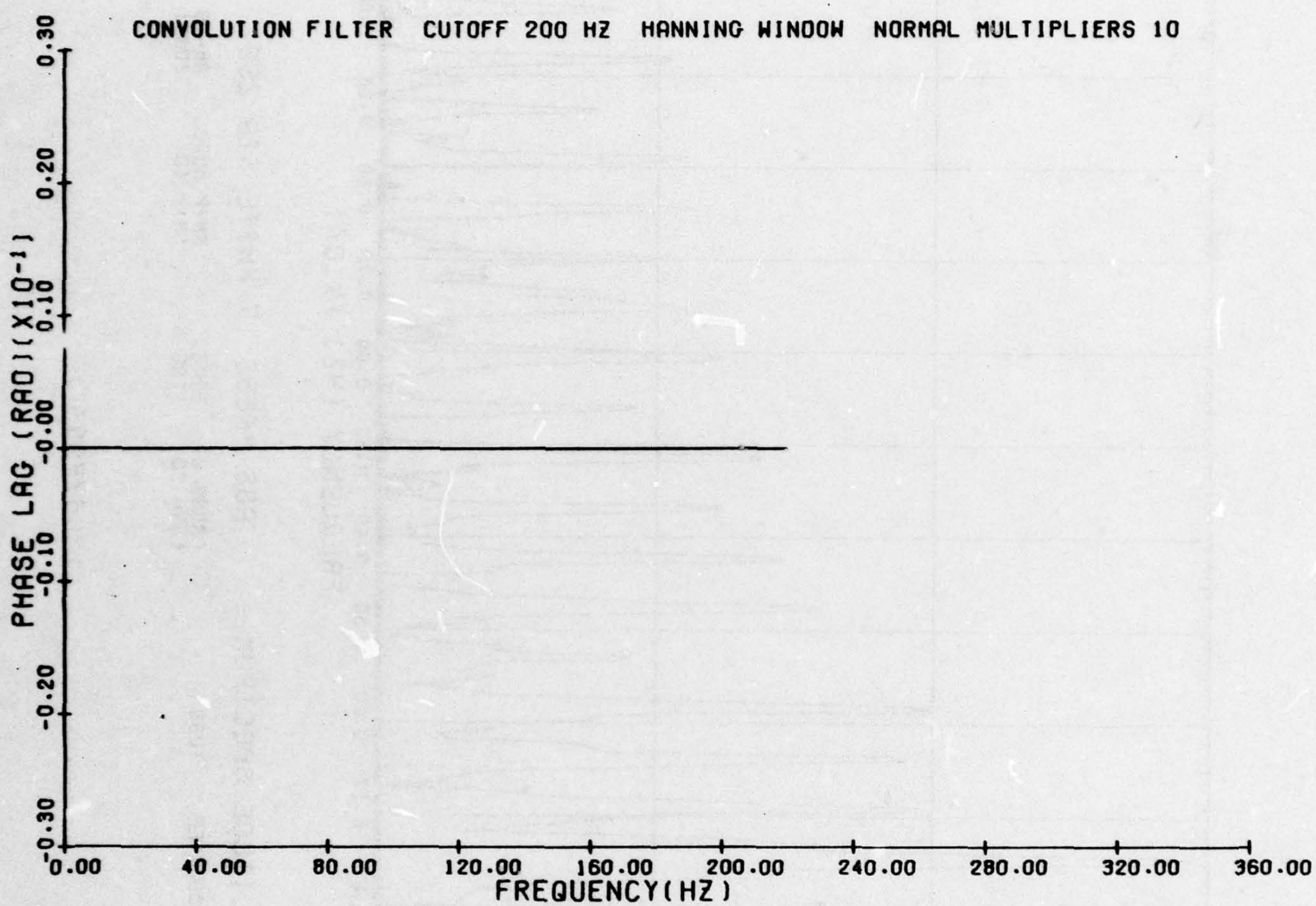
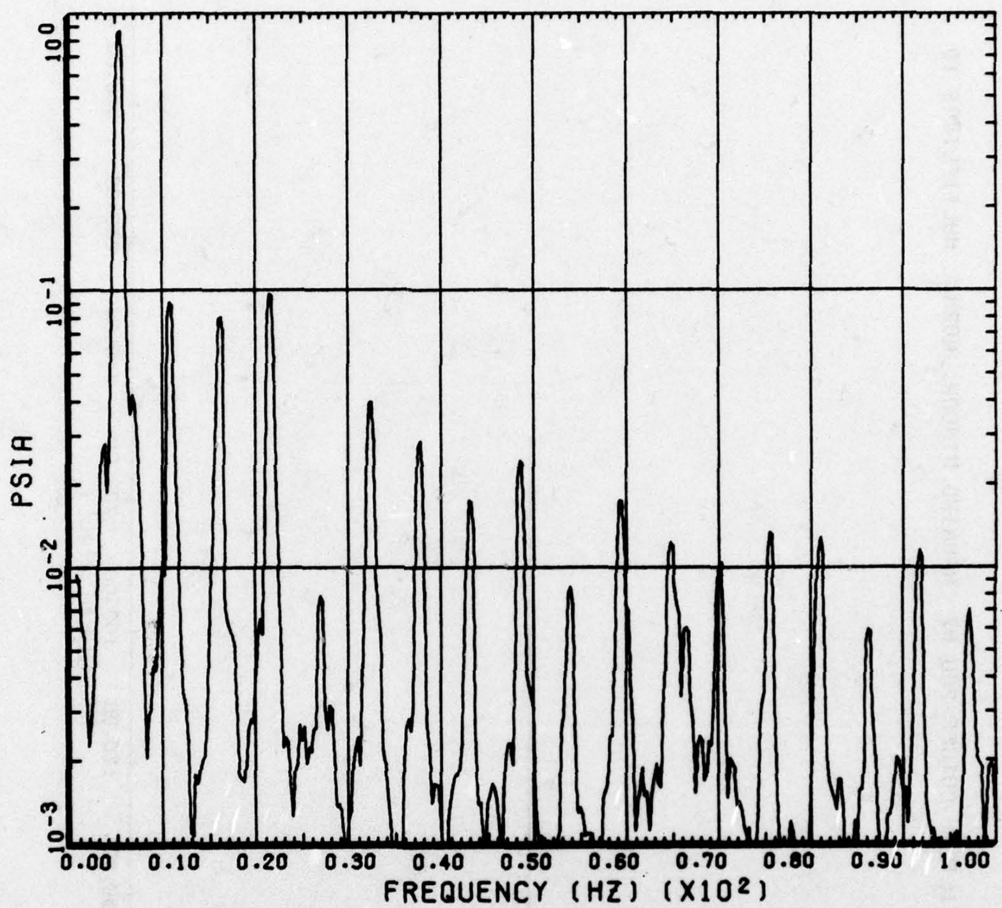


Figure 8. Convolution filter phase lag.



AMPLITUDE SPECTRUM - ABS PRESS 17 WHITE STA 252

COUNTER 1066 GROSS WT 8300. SHIP MODEL AH-1G
LONG CG 196.5 SHIP ID 20391

1066/P975

BHT.USARTL OLS/DMS (VERS 1.31 - 06/23/78) 07/31/78

Figure 9. Amplitude spectrum plot (log scale).

Harmonic Analysis computes the Fourier Series components for a user-specified integral number of rotor cycles. Output is amplitude and phase for each component versus harmonic number or frequency (Figure 10).

Digital Filtering will low-pass or band-pass filter a time history to eliminate undesirable frequency components. The user specifies the upper and lower break-frequencies and the number of poles in the mathematical representation of the filter. If the lower break-frequency is zero, the filter becomes a low-pass filter. Frequency components of the time history between the breakpoints in the pass band will be essentially unchanged in magnitude and phase. Frequency components outside the pass band will be greatly reduced in magnitude. Generally, more attenuation can be achieved outside the pass band by increasing the number of poles in the mathematical representation of the filter although three poles are usually sufficient. Filter users should be aware that the end point areas of time histories may be distorted by digital filtering. Usually, the problem is magnified by increasing the number of poles.

Moving Block Damping analysis assumes an input function of the form

$$F(t) = Ae^{-\frac{Dwt}{100}} \sin (wt + \phi) + Q(t),$$

where D is percent of critical damping, w is a known or suspected frequency component, and Q(t) is a function with frequency components well separated from w.

The output of moving block damping for a time history is the percent of critical damping, D, where a positive value implies the frequency component is stable.

Cycle Averaging takes a time history consisting of two or more complete rotor cycles along with a table of the end points of each cycle and averages together the cycles to obtain one representative cycle. The operation tends to reduce superfluous noise and reduces the effect of unique deviations in single rotor cycles.

Min/Max Analysis takes a time history consisting of one or more complete rotor cycles along with a table of the end points of each cycle and finds the maximum and minimum value occurring in each cycle. Oscillatory and mean values are calculated for each cycle where

HARMONIC ANALYSIS - ABS PRESS 2 WHITE STA 252

HARMONIC NUMBER	AMPLITUDE PSIA	PHASE DEGREES
0.100000E+01	0.590389E+00	-0.145646E+03
0.200000E+01	0.535133E+00	0.726258E+02
0.300000E+01	0.669266E+00	-0.139101E+03
0.400000E+01	0.620972E+00	-0.903367E+02
0.500000E+01	0.211360E+00	0.145936E+03
0.600000E+01	0.365225E+00	0.139632E+03
0.700000E+01	0.725495E-01	0.174187E+02
0.800000E+01	0.273794E+00	-0.227845E+01
0.900000E+01	0.122402E+00	-0.966605E+01
0.100000E+02	0.163188E+00	-0.133831E+03
0.110000E+02	0.137397E+00	-0.143559E+03
0.120000E+02	0.814845E-01	0.780913E+02
0.130000E+02	0.134617E+00	0.845891E+02
0.140000E+02	0.140932E-01	0.227448E+01
0.150000E+02	0.990313E-01	-0.531747E+02
0.160000E+02	0.257910E-01	-0.630331E+02
0.170000E+02	0.748642E-01	0.169811E+03
0.180000E+02	0.539417E-01	-0.177945E+03
0.190000E+02	0.318137E-01	0.321742E+02
0.200000E+02	0.403374E-01	0.385237E+02
0.210000E+02	0.146978E-01	-0.143664E+03
0.220000E+02	0.371723E-01	-0.119027E+03
0.230000E+02	0.383538E-02	-0.153609E+03
0.240000E+02	0.129437E-01	0.118660E+03

BHT.USARTL OLS/DMS (VERS 1.31 - 06/23/78) 06/30/78

Figure 10. Harmonic analysis printout.

$$\text{Oscillatory} = 1/2\ (\text{max} - \text{min})$$

$$\text{Mean} = 1/2\ (\text{max} + \text{min})$$

True Airspeed is derived from the Boom System Airspeed by smoothing, applying calibration constants, and correcting for outside air temperature (OAT) and static pressure. Boom System Airspeed is measured in knots squared and hence requires a square root operation.

Rotor Azimuth is estimated from values of time which mark the beginning and end of complete rotor cycles. These values of time are determined from the rotor azimuth pulse train and are corrected as required to account for offsets from zero degree azimuth.

Rotor RPM is the rotor speed in revolutions per minute (rpm) and is derived from the derived rotor azimuth.

Shaft Horsepower is the horsepower applied to the rotor as calculated from mast torque and rotor rpm.

Blade Static Pressure Coefficient (C_p) is the ratio of differential pressure at a blade station to the differential pressure expected on a flat blade with zero angle of attack produced by the motion of air parallel to the chordwise direction.

The Blade Normal Force Coefficient (C_n) is the integral around the airfoil of C_p components normal to the chordline. Radial position is held constant.

The Blade Chordwise Force Coefficient (C_c) is the integral around the airfoil of C_p components in the chordwise direction. Radial position is held constant.

The Blade Pitching Moment Coefficient (C_m) is the moment integral, with respect to quarter chord, of the C_p components in the chordwise and normal directions taken around the airfoil. Radial position is held constant.

Blade Displacement is the deformation in inches of the blade in the chordwise or beamwise direction for one user-specified harmonic (e.g., 2/rev). Blade displacement is calculated by

integration of the corresponding harmonic component of accelerometer measurements taken in the indicated directions. Prospective users of this derivation should be certain to consult Section 6.2 regarding deficiencies in this method of calculation of blade displacement.

Blade Slope is the derivative of the Blade Displacement with respect to radius and is derived from the representation of instantaneous Blade Displacement at several radial stations.

Blade Local Flow Magnitude is the magnitude of the velocity, in feet/second, of the air flowing over a Boundary Layer Button sensor pair. Blade local flow magnitude is calculated from the two perpendicular pressure measurements taken from a single Boundary Layer Button, Boom System static pressure, and OAT.

Blade Local Flow Direction is the direction in degrees of the airflow over a Boundary Layer Button sensor relative to the chordwise direction. Positive angles indicate air moving from outboard to inboard. Blade Local Flow Direction is calculated from the two perpendicular pressure measurements taken from a single Boundary Layer Button.

Shaft Thrust Coefficient is calculated from boom system static pressure, OAT, rotor speed, and vehicle weight, or in the case of the tail rotor, antitorque force.

Shaft Torque Coefficient is derived from mast torque, boom system static pressure, OAT, and rotor speed.

Density Altitude is the altitude above sea level corrected for density variation with altitude. Density altitude is computed from OAT and Boom System static pressure.

2.5 DIMENSIONAL CAPABILITIES

Many input or output data streams specified by the user will be simple functions of time, azimuth, frequency or harmonic number. However, the processing program is also designed to accommodate data which may be expressed as a function of several independent variables. These variables are referred to as dimensions. The first dimension is always time or a time-related variable since the system is designed to process time histories. Frequency and harmonic number are time-related variables. When an azimuth record is available, azimuth can be associated with time for purposes of input definition, processing, or display.

Two other dimensions can be handled by the program. The second dimension and frequently the third dimension are defined from the positions of multiple sensors which measure like parameters. For example, the position of several microphones on the ground which formed a line perpendicular to the direction of travel of a helicopter could be either the second or third dimension.

The independent variable could be distance from the microphones to the intersection of the line and the helicopter path. The specification of dimensional organization of sensors which measure like parameters is performed by the Info File (see Section 4.8).

The second dimension is frequently referred to as the row position dimension. For the OLS blade application, the second (or row) dimension is the chord position. Positions in the chordwise direction are expressed as fractions of the chord length (e.g., .70 X/chord).

The third dimension is also called the column position dimension. This dimension can have either a geometric or multi-counter interpretation. For the OLS blade application with the geometric interpretation, the third or column dimension is associated with radial position on the blade. Positions in the radial direction are expressed as fractions of the blade radius (e.g., .95 R/RADIUS).

As indicated, the third dimension could also refer to multiple counters where each column element corresponds to a different counter. For example, the program could process data from all the absolute pressure sensors at one radial station for several different counters. The output could then be displayed as a function of chord position and airspeed. Scratch files (Section 2.6) must be used to realize this capability. Multiple counter processing is used to generate a significant True Airspeed or rpm variation for plotting when these parameters are relatively constant during any one counter.

The input and output data specified need not have 'extent' in all of the allowed dimensions. Frequently, for single functions of time, azimuth, on a time related variable, multiple dimension specifications can be ignored. Occasionally, as with blade local flow or blade static pressure coefficient derivations, row and column element specifications are required even though a single blade station is to be processed. Another example of zero extent in a dimension is the selection of a single time instant or azimuth position for display or processing of data from multiple row and column positions.

Frequently, the program must process two time histories for each row/column intersection. For example, considering the group of absolute pressure sensors on the OLS blade, for every chord/radius intersection there are two sensors; one sensor on the top surface and one sensor on the bottom surface. Time histories from each of these two sensors are referred to by the program as double-row elements. Data from the top surface sensor is referred to as the 'top' double-row element and data from the bottom surface sensor is referred to as the 'bottom' double-row element. The use of double-row elements is not restricted to the above example. For example, boundary layer

button (BLB) data from the inboard pointing BLB sensor is processed as the 'top' double-row element and data from the outboard pointing BLB sensor is processed as the 'bottom' double-row element.

Either one or both double-row elements may be present in processing. The number of elements on input need not correspond to the number on output. For example, Harmonic Analysis takes one double-row element time history and creates the two double-row elements amplitude and phase for output. When no double-row context is present, the 'top' double-row element is presumed in processing and the user need not consider double-rows.

The information required to specify the sensors and geometric positions of the sensors for these multiple dimensions need not be entered by the user while executing the processing program. These data are contained in the information file or Info File. The user need only specify the group (a subset of the Info File providing data on a particular kind of sensor) in the Info File along with the dimensional elements desired. For example, in the OLS context, suppose a user wishes to perform a process with all the absolute pressure sensors at .40 R/RADIUS (40% radius) radial position on the blade, the user should specify the Info File group which contains the blade absolute pressure sensors. In addition to the group specification, the user should also specify element one of the third dimension (radius variable), all elements of the second dimension (chord variable), and both double-row elements (top and bottom surfaces). All of the appropriate sensors, along with their respective geometric positions, are identified in the Info File. Refer to Section 4.8 for information on the structure and syntax of the Info File.

2.6 SCRATCH FILES

Rather than printing or plotting the results of any process, the user may choose to have the Processing Program store the output on a scratch file. A scratch file is a disc storage area which is not maintained after a run of the processing program. Two scratch files are visible to and accessible by the user and are referenced as SCF1 and SCF2. Either file will hold data organized in all the dimensions of processing listed in Section 2.5 within the limits of available storage.

Once a process result has been stored on a scratch file, that processed data can be used as input for a new process or simply used for display. All or part of the scratch file contents can be accessed at the user's option.

One of the primary advantages provided by scratch files is the ability to perform sequences of processes on the same data. An example is outlined in Figure 11. A user starts with absolute pressure data for all radius and chord positions on the top and

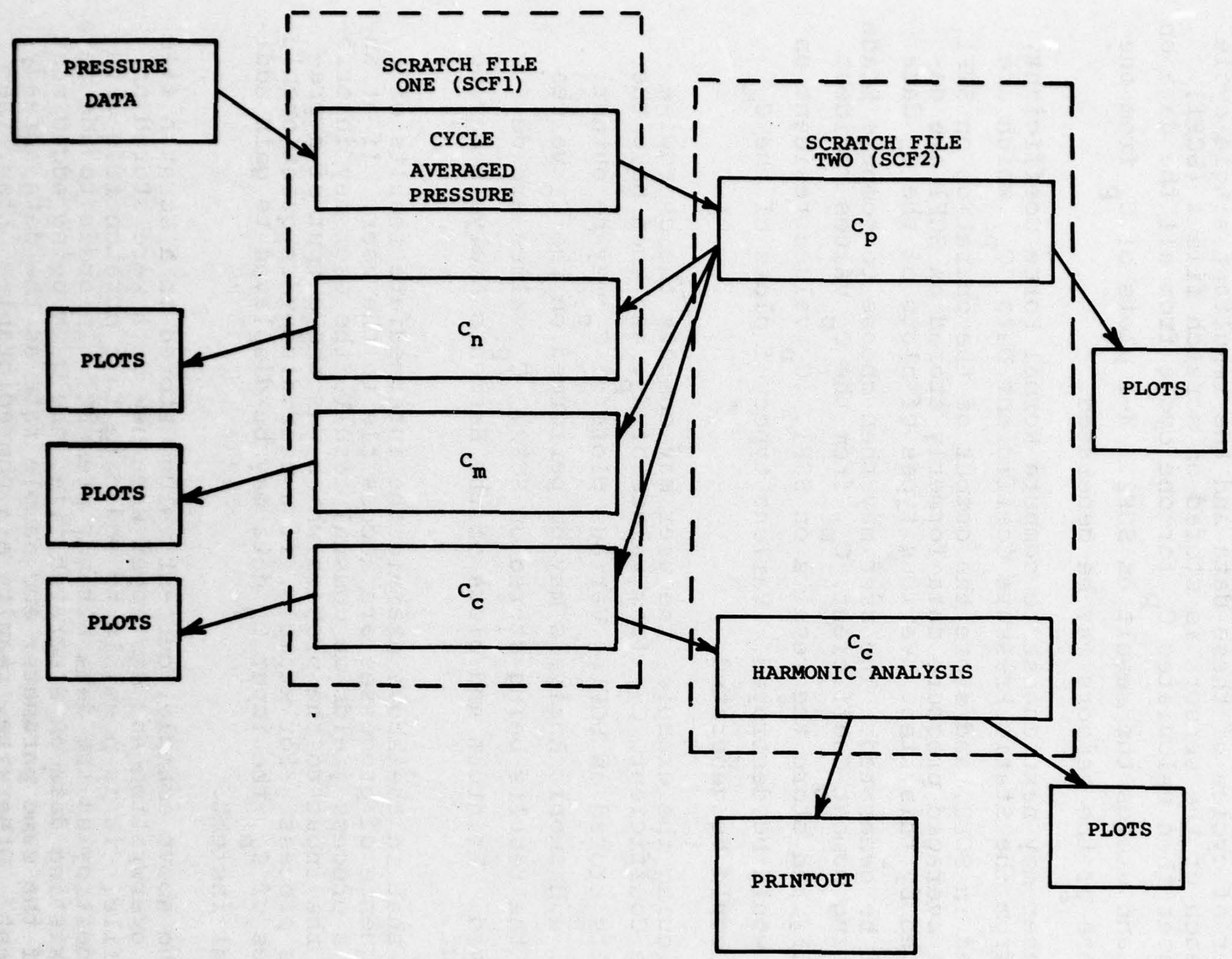


Figure 11. Linking processes with scratch files.

bottom surfaces of the blade which are loaded in a partition of the Master File. Cycle averaging is performed on some integer number of cycles of these data and the resultant single cycle for each of the sensors is stored on scratch file 1 (SCF1). The user then calculates C_p for one cycle from all the data on SCF1 and stores the result on SCF2. X-Y plots of C_p from one or more of the sensors may be generated.

The user may next choose to compute Normal Force Coefficient, C_n, from the Static Pressure Coefficient Data, C_p, which are stored in SCF2, and store the output of the operation on SCF1. Cycle averaged pressure data formerly stored on SCF1 are destroyed by this step. Various types of plots of the C_n data could be generated. The user may then choose to compute Blade Pitching Moment Coefficient, C_m, from the C_p values resident on SCF2 and store the results on SCF1. C_n values resident on SCF1 would be destroyed. Various types of plots of the C_m data could be generated.

Continuing the example, the user may compute the chordwise Force Coefficient, C_c, from values of C_p, and then have the results stored on SCF1. Various plots of C_c may be output. Next, a Harmonic Analysis may be performed on the C_c values with the results being stored on SCF2. C_p values are destroyed. Printout and plots of the Harmonic Analysis may be output.

Note that in the above example the intermediate results of a sequence of processes are accessible to the user. If at any time a process yields an unusual result, the user may interrogate the input to the process which was the output of a previous process. For example, if a user discovers questionable values of C_n, the input C_p data may be displayed to gain additional insight.

In the above example, current values stored in a scratch file were overwritten and destroyed when new data were stored on the file. It is possible to add data to a scratch file without destroying the data already present. In order to add data to existing data on a scratch file, the data being added must be of the same parameter and sample rate as the data already present. Otherwise, results are unpredictable. Also, the data being added can represent only one column (radius) position although it can have multiple row (chord) positions and double-row elements. This restriction is present because each ADD to a scratch file is considered to be a third-dimension element or column.

The capability to ADD data to a scratch file is useful when generating plots which contain data of more than a single counter. If the user wished to compute Shaft Torque Coefficient as a function of airspeed, data from several counters with differing airspeeds but otherwise like conditions could be used. Shaft Torque Coefficient may be computed for each counter separately and the results added to SCF1. Plots of Shaft Torque Coefficient versus airspeed may then be generated.

2.7 ATTACHED PARAMETERS

Certain variables are required very frequently for analyses, derivations and displays. These variables are called attached parameters. Attached parameters include rotor azimuth, true airspeed, rotor rpm, Outside Air Temperature, and Boom System static pressure. Attached parameters are assumed to be slowly varying and are smoothed in derivation and stored at the rate of one sample per rotor cycle.

Attached parameters are calculated whenever required for a process, a plot scale, or when the output is to a scratch file. The values are maintained in the memory of the computer so that they need not be recalculated when a new step references the same counter and time period. When results are written to a scratch file, all of the attached parameters are also stored. These parameters are then available for subsequent analyses, derivations, or plots involving the stored results.

Manipulation of the attached parameters is transparent to the user with two exceptions. The user must assure that the proper item codes are present on the Master File partition to provide for the derivation of rotor azimuth, indicated airspeed, outside air temperature and Boom System static pressure. Alternatively, the user has the ability to specify, for some derivations, a constant value for OAT and/or static pressure.

2.8 OUTPUT CAPABILITIES

Process results or raw data can be displayed by the Processing Program in two basic forms: printout and plots. Printout consists of tabular listings of one or two double-row elements along with time, or a time-related variable. For outputs with more than one second or third-dimension element, each row/column intersection forms a new tabular listing. Listings generated in the batch mode of operation are routed to a line printer, while listings generated in the interactive or interactive graphics mode are routed to the interactive terminal.

Four general kinds of plots are available: simple X-Y plots, multiple-curve X-Y plots, surface plots, and contour plots. Simple X-Y plots depict a single function of one variable with vertical-dependent variable and horizontal-independent variable axes. The increment between annotated points on each axis is restricted to powers of ten multiplied by the integers 1, 2, 4, or 5. Figure 12 shows an example of a simple X-Y plot.

The user has the option to specify a logarithmic scale for the independent and/or dependent variable of an X-Y plot. With a log scale specification, the user must indicate the number of decades to depict for the axis in question. The highest decade on an axis will always be the one which includes the largest value present for the corresponding variable. Zero and negative values which occur in a variable to be plotted with a log scale are reset to a very small positive number. Figure 9 provides an example of a log scale.

Linear independent and dependent variable scales are normally scaled automatically to include all values on the plot. However, the user has the option to specify either scale size by specifying the minimum value present and the increment between annotated points. Nine increments, or divisions, are provided on the vertical axis, while ten increments are provided on the horizontal axis. The increment must be a power of ten, which is multiplied by one of the integers 1, 2, 4 or 5. Similarly, the minimum value must be a multiple of the increment. Values in the function which exceed the user-specified scale will be clipped. Figure 13 shows a plot with a user-defined scale.

When X-Y plots are to be drawn on the Tektronix 4014, the user may specify that the crosshair cursor be activated immediately after the plot is completed. The crosshairs may be used to evaluate points on the screen in user coordinates.

Multiple-curve X-Y plots are essentially the same as simple X-Y plots except that more than one function may be shown on the same plot. Curves on the plot are differentiated by various sequences of dashed lines. Samples of the dashed lines are labeled below the plot. Data plotted on multiple curve X-Y plots should be of like kind and similar in scale. Figure 1 provides an example of a multiple-curve X-Y plot.

In addition to the multiple curve plots described above, a facility exists to add curves to an existing plot. New variables should agree in kind with the variables shown on the original plot scale. The user should plan ahead in specifying the scale parameters so that subsequently added curves

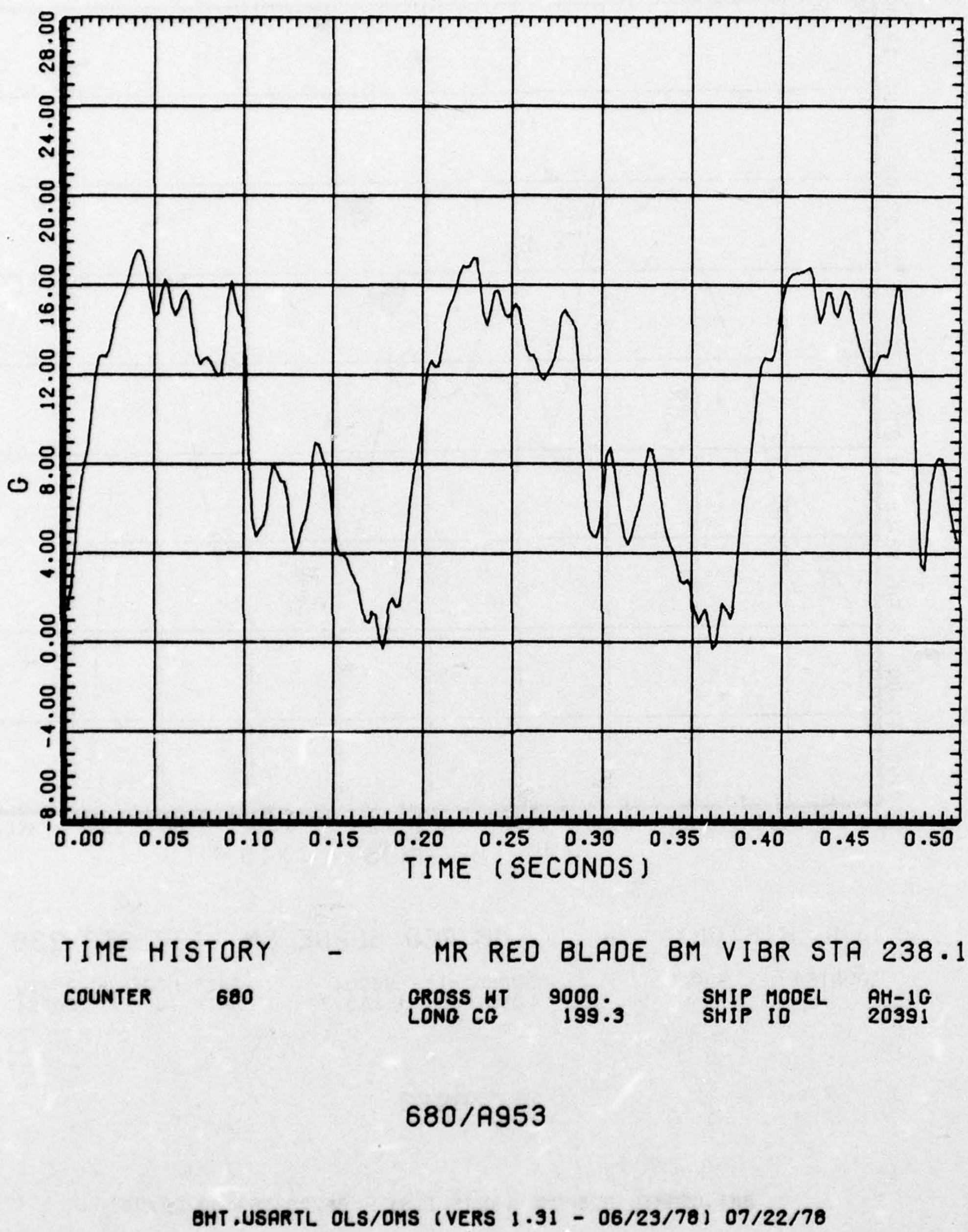
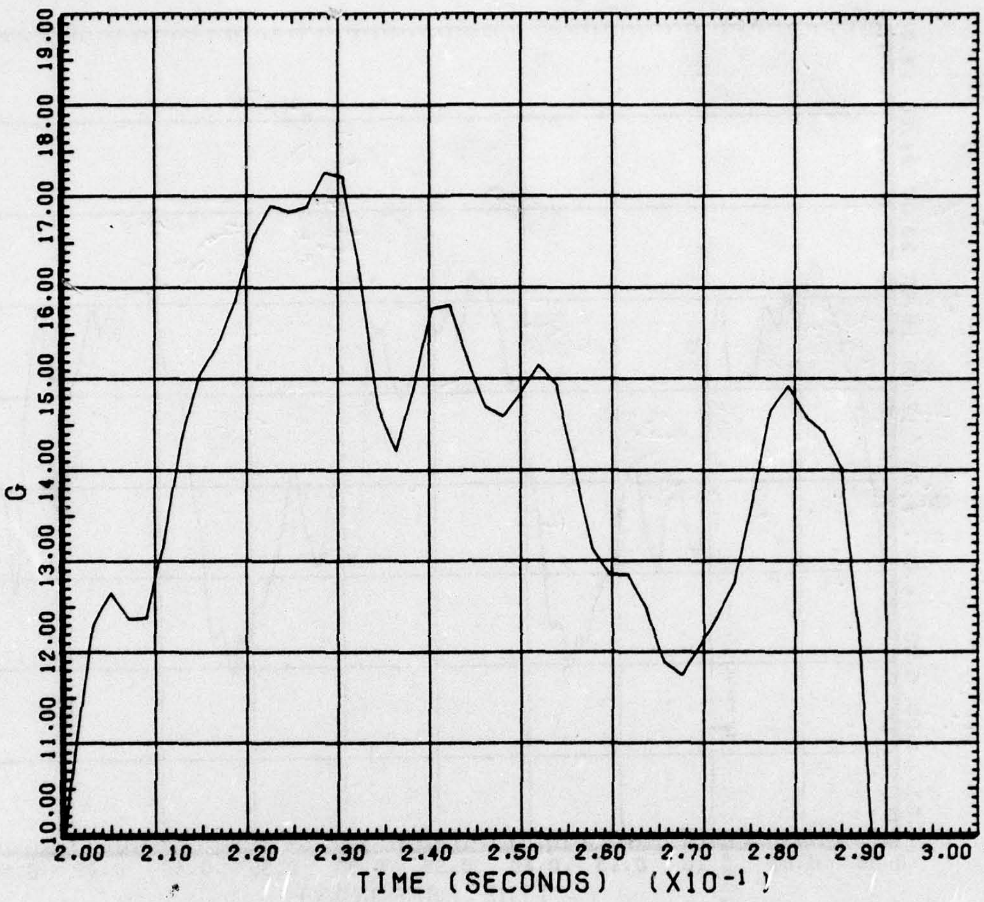


Figure 12. A simple X-Y plot.



TIME HISTORY - MR RED BLADE BM VIBR STA 238.1

COUNTER	680	GROSS WT	9000.	SHIP MODEL	AH-1G
		LONG CG	199.3	SHIP ID	20391

680/A953

BHT.USARTL OLS/DMS (VERS 1.31 - 06/23/78) 07/22/78

Figure 13. An X-Y plot with user defined scale.

are not clipped since automatic scaling cannot account for the magnitude of subsequent curves.

Surface plots depict a function of two independent variables in some three-dimensional space. The user is provided a perspective view of the surface which is represented by a set of curves along each independent axis. Each set of lines is equally spaced and orthogonal to the other independent axis. The coordinate system for surface plots can be rectangular or cylindrical. For cylindrical plots, one independent variable must be a complete cycle of rotor azimuth. Figure 4 shows an example of a surface plot in cylindrical coordinates, while Figure 14 shows a surface plot in rectangular coordinates.

Algorithms used to formulate surface plots permit the user to observe the surface from essentially any point in that three-dimensional space. The position of the observer in 'X', 'Y', and 'Z' is specified in units of the maximum width of the surface. The observation point must not be located inside the object. Observation points located within one object width from the center create an apparent distortion in the plot similar to apparent distortions in a photograph when the camera is held very close to the subject.

Contour plots depict a function of two independent variables by drawing lines of constant dependent variable values. These lines are differentiated for the user by the following scheme. Contour levels are equally spaced and the contour lines for every fourth level are solid with letters superimposed on the curves. Samples of these lines are drawn below the plot and labeled with the function value. Three intermediate contour levels are represented by dashed lines. The length of the dashes decreases with increasing contour level. Contour plots can be drawn in rectangular or cylindrical coordinate systems. For cylindrical plots, one independent variable must be a complete cycle of rotor azimuth. Figure 2 shows a contour plot in rectangular coordinates, while Figure 3 shows a contour plot in cylindrical coordinates.

For contour plots, the user has the option of specifying the minimum contour level of interest and the increment between levels. Normally, the minimum contour level and the increment between levels are selected by an algorithm similar to the method used for selecting scale values on X-Y plots. Contour level values are always 1, 2, 4 or 5 multipled by a power of 10. The user may specify a minimum value and increment which is consistent with these rules but which may increase (or decrease) the number of levels for some range of contour values.

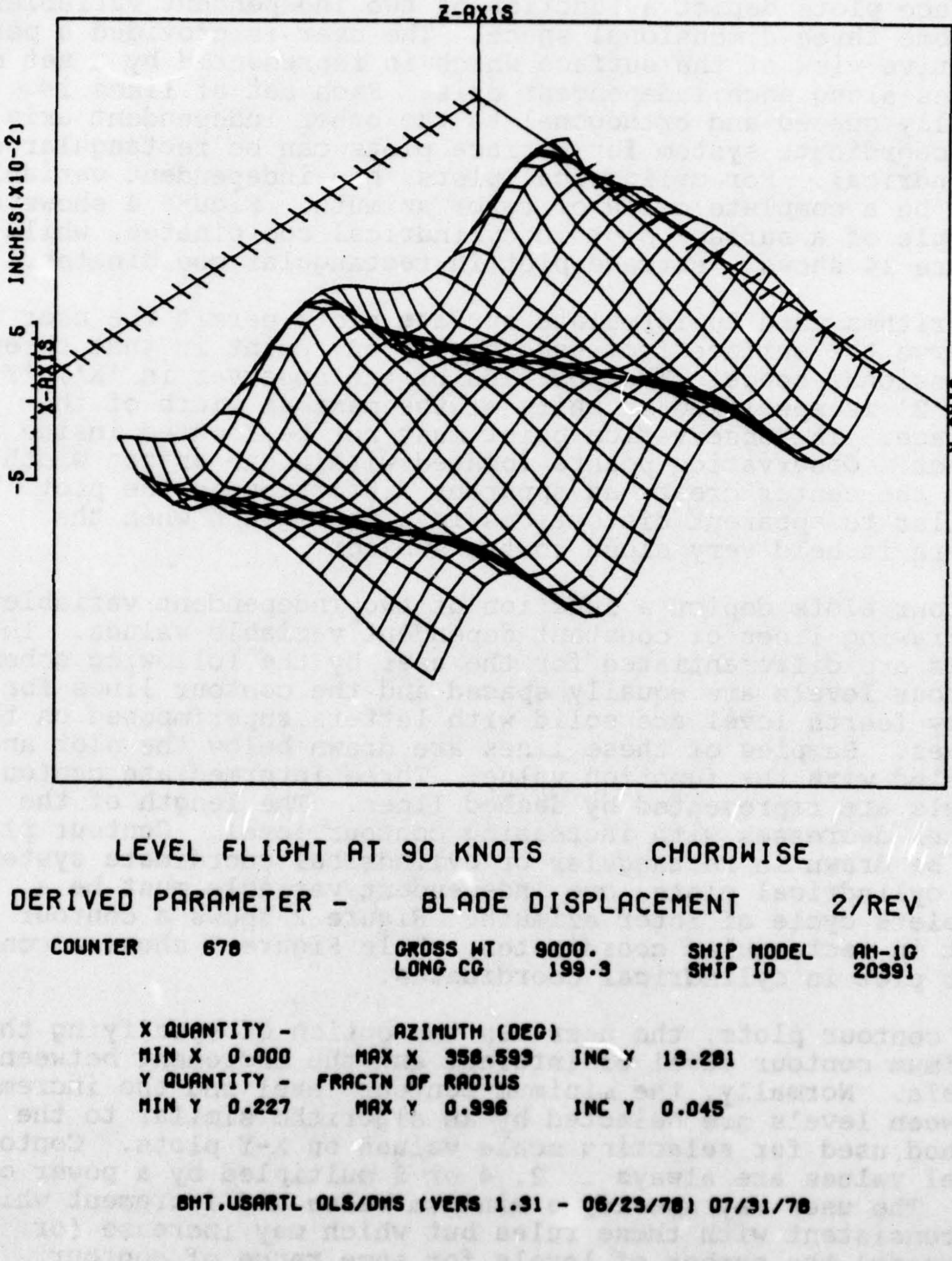


Figure 14. Surface plot in rectangular coordinates.

For incremental plotter output, the graphs are designed to fit on preperforated 8-1/2 by 11-inch paper. Each plot frame is translated 8-1/2 inches to the right of the previous one. This 8-1/2-inch-frame increment may be modified by the user at the beginning of a run of the Processing Program.

For Tektronix output, the plots are drawn on the right side of the screen and occupy approximately 60 percent of the screen width. Prompting messages from the program and inputs from the user are written on the left-hand side of the screen (Figure 15). Tektronix plots are nearly identical in size and form to corresponding incremental plotter graphics.

```
NEW STEP   STEP CPU   0.01 SEC   TOTAL CPU   21.0 SEC
DISP/SCF1 45 MRAZ/APLOT/
 EXECUTING

NEW STEP   STEP CPU   0.01 SEC   TOTAL CPU   21.2 SEC
DISP/SCF1 50 MRAZ/APLOT/
 EXECUTING

NEW STEP   STEP CPU   0.01 SEC   TOTAL CPU   21.4 SEC
DISP/SCF1 55 MRAZ/APLOT/
 EXECUTING

NEW STEP   STEP CPU   0.01 SEC   TOTAL CPU   21.6 SEC
DISP/SCF1 60 MRAZ/APLOT/
 EXECUTING

NEW STEP   STEP CPU   0.00 SEC   TOTAL CPU   21.9 SEC
DISP/SCF1 65 MRAZ/APLOT/
 EXECUTING

NEW STEP   STEP CPU   0.01 SEC   TOTAL CPU   22.1 SEC
DISP/SCF1 70 MRAZ/APLOT/
 EXECUTING

NEW STEP   STEP CPU   0.01 SEC   TOTAL CPU   22.4 SEC
DISP/SCF1 75 MRAZ/APLOT/
 EXECUTING

NEW STEP   STEP CPU   0.01 SEC   TOTAL CPU   22.6 SEC
DISP/SCF1 80 MRAZ/APLOT/
 EXECUTING

NEW STEP   STEP CPU   0.01 SEC   TOTAL CPU   22.8 SEC
DISP/SCF1 85 MRAZ/APLOT/
 EXECUTING

NEW STEP   STEP CPU   0.00 SEC   TOTAL CPU   23.0 SEC
```

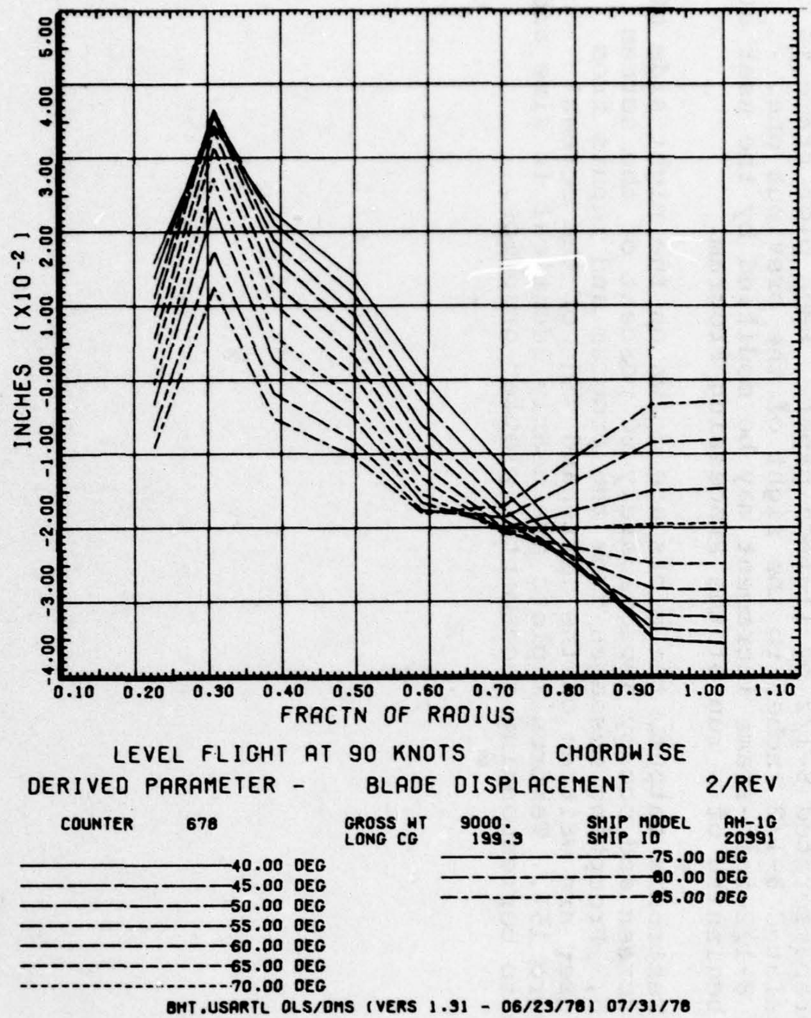


Figure 15. Tektronix screen apportionment.

3. DATA FILE CREATION/MODIFICATION

3.1 FILE INITIALIZATION

Assuming that the space for a Master File exists on a direct access device (i.e., disc), it must first be initialized before data can be stored on it. Specifically, for IBM system application, the initialization process is to write zeroed records on every record position in the direct access file. If an initialization capability were included in the File Creation Program as an option, a user might inadvertently initialize the Master File when it contained useful data.

The Master File Initialization Program performs the function of writing on every record in the file. This program then presets the first, or directory record, for the Master File with the following information: the total number of records in the file, a password for the entire file, and a list of partitions showing that none are present. This program requires one line of user input in free-field format containing two entries which may appear in either order. One entry is an integer specifying the size of the Master File in records (currently 256 words or 1024 bytes per record). The other entry defines the password for the entire Master File called the superword. The purpose of the superword is explained in Section 3.4.

If the disc space allocated for the Master File is insufficient for the number of records specified by the user, the program will terminate abnormally when it attempts to write on a record which it cannot access. While running, the program lists every 50th record it is attempting to write on so that, in the event of an abnormal termination, the user will have an estimate of how many records were successfully written. If the intialization is successful, the program will printout the message (for example):

```
FILE INITIALIZATION SUCCESSFUL
RECORDS = 20800
SUPERWORD = BIGWORD
```

The user can then proceed to submit batch runs of the File Creation Program to transfer data to the Master File.

3.2 FILE ADDITIONS

3.2.1 Characteristics of the File Creation Program

The File Creation Program transfers data from digital tape (or some other storage medium) to the Master File. Data selection for transfer and control parameters for this program is supplied by user input, which must be in a format explained in Paragraph 3.2.3. This input specifies item codes and counters to be transferred along with other information.

The program attempts to transfer all the item code/counter combinations for all the item codes and counters specified. If an item code/counter pair is not found in the available input data, the corresponding data are simply left off the Master File partition. Data which are found are still written on the partition even though other requested data cannot be located on the input medium. Sometimes the amount of data available for a specific item code/counter is less than the time history length requested. In this case, the truncated data record is transferred.

When counters are specified, a time history length and offset from the start of input data to the start of the time histories to be transferred is associated with each counter. This offset and length applies to every time history (item code/counter pair) corresponding to that counter. Each item code specified has associated a sample rate reduction factor, a digital filter breakpoint (-1.0 implies no filtering), and a format for data storage on the Master File (calibrated or uncalibrated). These factors will apply to every time history corresponding to that item code.

The File Creation Program can operate in three different modes with regard to the partition which is receiving the data. In the NEW mode, a new partition with a new partition name is created. In the ADD mode, data are transferred to an existing partition without disturbing data already present except that duplicate item code/counter pair time histories are overwritten. In the REPLACE mode the data on an existing partition are deleted and new data are written into the partition.

In the NEW mode, a password is set for the partition. The same password must be entered in any subsequent run of the File Creation Program using the ADD or REPLACE mode for the same partition. The intent of this security feature is to prevent inadvertent rather than intentional destruction of data. The Master File monitor (see Section 3.5) has the capability to override the password, using the superword capability.

After all the data specified in the user input have been transferred, or after the data input medium (digital tape) is exhausted, the File Creation Program can provide two services for the user. The first of these services is the generation of a Map, or listing, of the contents of the partition. The Map contains a list of counters present and, for each counter, a list of the item codes present. For each item code, the following information is included: record length, sample rate, digital filter applied (-1 indicates no filtering), specified offset, time alignment offset, item description, and item units. If the user wishes to suppress the Map, then the instruction 'NO MAP' must be specified.

The second post-transfer service that the File Creation Program can perform is a SAVE of the partition. When the user specifies SAVE in the input, the program will copy the partition to digital tape. The SAVE process is the same as the partition SAVE used in the File Maintenance Program (see Section 3.4). The user has the responsibility to assure that the partition will fit on one digital tape (see Section 3.4). Restoration of a partition copied in this fashion must be performed with the File Maintenance Program. The user must specify SAVE in the input when a partition copy is required since SAVE is not a default entry.

Computer CPU time limit settings are critical in the operation of the program. If the program runs out of time before transfer of data to a partition is complete, then that partition will certainly be lost. In some cases, the entire Master File could be destroyed if a time limit specified were too short. The user must assure that a generous time allotment is specified for each run of the File Creation Program when data already exist on the Master File.

Unfortunately, it is impossible to provide the user with a precise method for estimating the computer CPU requirement for a run. Time requirements can change considerably as computer CPU models and local installation factors are changed. Time requirements can also change as processing factors (e.g., digital filtering) are altered. For example, on the IBM 370/168 installed at BHT, about five minutes were required to create a partition of 15,000 records from 16 digital tapes using no digital filtering. Digital filtering might have increased the time requirement substantially. Sample rate reduction would reduce the filtering time and also reduce the size of the partition. If the total Master File space was 30,000 records of which 15,000 were already occupied, then an additional zero to five minutes could be required to rearrange the partitions in the Master File to create the maximum space for the new partition. The user must gain experience with the CPU time usage for this program while being extremely generous in time limit specifications.

3.2.2 Setup Actions Required

Before attempting to build an input data set for the File Creation Program, the user must gather certain information. The item codes and counters which specify the data necessary for processing must be determined. The user must then find out which digital tapes contain these data and the sample rate for each time history to be transferred. For each counter selected, the user must determine the record length for the time histories to be transferred, and a time offset between the start of data on tape and the beginning of the data to be transferred. The time offset may be zero or any positive time interval which is less than the length of the corresponding time histories stored on tape. The record length for the time histories to be transferred is bounded by the length of the time histories on tape less any time offset applied.

For each item code, the user must calculate the data rate reduction factor to be applied and decide whether the data should be calibrated and/or filtered (filtered data must be calibrated). If, for example, the user wishes to perform calculations using multiple sensors of the same type (e.g., blade absolute pressure sensors to calculate C_n), then the user must assure that the data rate reduction factors applied to the time histories from all these sensors result in identical sample rates for all the corresponding time histories stored on the Master File.

The user should then estimate the number of records on the Master File required to store these data. Four kinds of records are required to store these data: data storage records, time history information records, partition directory records, and partition records. Eight partition records are always required. The following is the formula for determining the number of records required for storage of data on the Master File.

$$A = N \times M$$

$$B = \sum_{i=1}^{N} \sum_{j=1}^{M} [R_i L_j / K]$$

$$C = [M/128] + M \times [N/128]$$

$$S = 8 + A + B + C$$

where

N = number of item codes

M = number of counters

R_i = Final sample rate for item code "i".

L_j = Final record length for counter "j".

K = 256 if data are calibrated or 512 if data are uncalibrated.

S = number of Master File records required for storage of the data.

This formula is valid only for the initial storage of data on a partition, or for a complete requirement of a partition.

This estimation procedure is provided for use when Master File space is restricted. If space is plentiful, then a rough estimate of space required should be adequate. Rough estimates should be generous since the File Creation Program will not exceed the specified partition size (SPACE entry). When all the requested space is not used, the partition produced will include only the records necessary to store the data. Any surplus storage area is returned to the available pool.

3.2.3 User Instructions

Input to the File Creation Program will consist of a sequence of input lines of up to sixty characters, each line containing a number of entries. An entry is a sequence of alphanumeric characters unbroken by commas or blanks. A comma or any number of blanks separates the entries. Input is regarded as a continuous sequence of entries from the first entry on the first line to the last entry on the last line. However, a single entry cannot span two lines.

There are two broad categories of entries, literal and numeric. The first character of a numeric entry is '+', '-', '0-9' or '.'. A numeric entry is free field and is interpreted as integer or floating according to context. Thus, decimal points are not required for floating input unless necessary to include a fractional part. Exponential entries (e.g., .1260E-5) are not allowed. Literal entries begin with any characters other than those which begin numbers and are interpreted as a character string (e.g., FILTER, JOHN-SMITH, PRESSDAT).

The first kind of literal is an 'instruction word'. An alphabetical list of instruction words appears with individual explanation later. Instruction words can specify instructions by themselves or may require a qualifying string or numeric entry to complete the instruction. Instruction words have a preassigned meaning which control the input.

A second kind of literal is the qualifying string mentioned above, e.g.,

NEW PRESSDAT

'PRESSDAT' would be a data set name and is the qualifying string.

The final kind of literal is an item code. Any literal which is not an instruction word or qualifying string and which is exactly four characters long is assumed to be an item code.

The first kind of numeric input is a qualifying number, a value which must go with an instruction word. For example,

SKIP 32

means record only every 32nd data point read from tape. A number which is not a qualifying number is assumed to be a counter if it is in the range 1-32767.

A description of each instruction string follows. Note that initial letters of each word are underlined. These are the minimum number of letters required to define the word. Additional letters up to a total of four must match; others are ignored. Thus: CA means the same as CALIBRATE which could also be written: CALIXXX, CALI or CAL but not CALX.

<u>AB</u>SOLUTE (number)	Absolute offset. 'Number' seconds of data (floating) will be discarded starting with the first point found regardless of time skew alignment requirements. This value is associated with all counters input after the instruction until a new ABSOLUTE or OFFSET command is found.
<u>AD</u>D (literal)	Add data to a partition. The data to be input will be added to a partition of data already present with name identical to the literal.

ALIGN	System will attempt to align first points recorded so that every item code begins at the same time. This instruction may appear anywhere before the END instruction except where it would affect the context of other instructions.
CALIBRATE	Store calibrated data. Data for all item codes which appear after this instruction and before any subsequent NOCALIBRATE command will be stored calibrated.
COUNTERS	A comment command to indicate that a block of counters will be input. This instruction is ignored by the software.
END	End of input. Any entries which come after the END will be ignored.
FILTER (number)	Low-pass filter breakpoint. Indicate the upper breakpoint in hertz for a filter to be applied to every item code which appears after this instruction and before a subsequent FILTER or NOFILTER instruction. Data will be calibrated regardless of CAL or NOCAL instructions.
ITEMS	A comment command to indicate that a block of item codes will be input. This instruction is ignored by the software.
NEW (literal)	New partition. The data to be input will be a new partition with name given by the literal. The partition name may include from 1 to 8 characters.
NOFILTER	No filter will be applied to data from item codes which appear after this instruction and before a subsequent FILTER instruction.
NOCALIBRATE	No calibration will be applied to data from item codes which appear after this instruction and before a subsequent CALIBRATE instruction.
NOMAP	After partition generation or modification is complete, a list of item code-counter pairs present in the partition is normally generated. NOMAP suppresses this list.

OFFSET (number)	Offset from aligned value. Offset 'number' seconds (floating) after alignment. Equivalent to ABSOLUTE if ALIGN is not specified.
PASSWORD (literal)	If NEW, enter (literal) as the password for the partition. If ADD or REPLACE, enter (literal) to allow the modification of the partition. The password may include from 1 to 16 characters.
RECORD (number)	Amount of data read. System will attempt to read 'number' seconds of data (floating) for the counters which appear after this instruction and before a subsequent RECORD instruction. The number of points transferred depends on the SKIP factor (below) and the sample rate for the data as stored on tape.
REPLACE (literal)	Replace a partition. The partition (literal) will be deleted and a new partition with the same name will be built.
SAVE	After partition generation/modification is complete, this command will generate a tape save of the partition (the tape must be specified in the JCL-FT15F001)
SKIP (number)	Skip factor. Reduce the number of points transferred by using only every 'number' point in the sequence. A skip factor of one means no skipping. Applies to item codes which appear after this instruction and before a subsequent SKIP instruction.
SPACE (number)	Storage space allowed for partition. Number of random access records (1024 bytes/record) to be allowed as maximum size for partition. For an ADD, this limit includes the space already used by the partition before additions.
STRANGE	Specifies that the data input tape format will be other than the standard BHT format.
TAPES (number)	Number of data tapes to be scanned for data.

THRU (number)	Requires counters to be entered before and after this entry and specifies that all counters between those entered before and after the 'THRU' entry will be input. Thus, 28 THRU 32 means counters 28, 29, 30, 31, and 32 will be used.
USER (literal)	Enter user name. (Literal) will be substituted as the partition user name if the password is correct. The user name may include from 1 to 16 characters with no blanks.

ORGANIZED EXAMPLE -

```
NEW BLADESET USER J-SMITH PASS ABCD
ALIGN SPACE 2000 TAPES 2
ITEMS
CAL SKIP 4
R123 R124
NOCAL SKIP 8
R281 R282 R283
FILTER 16 SKIP 32
R242 R243 R244
COUNTERS
OFFSET .25 RECORD 2.0
500 512 581 616 THRU 620
OFFSET 0 RECORD .5
782 783
ABS 1.2 RECORD 1.0
10 12
END
```

DISORGANIZED EXAMPLE
(Will give same result)

```
CA SK 4 R123 R124 NOC SK 8
R281 R282 R283
FIL 16 SK 32 R242 R243 R244
NEW BLADESET PASS ABCD US J-SMITH
OFF .25 REC 2.0 500 512 581 616 THRU 620
OFF 0 REC .5 782 783 ABS 1.2 REC 1.0
10 12 SPAC 2000 TA 2 ALI END
```

3.2.4 Data Rate Reduction and Filtering Considerations

Section 2.2 included short discussion of the possible need for low-pass filtering when the sample rate for data is reduced. This section will give some guidelines for use of low-pass filters to avoid aliasing problems and to remove noise.

Figure 7 shows the magnitude part of the 'transfer function' of the low-pass digital filter used in the File Creation Program when a breakpoint of 200 Hz is specified. This means that a Fourier component in the input time history would be multiplied by the corresponding magnitude part of the transfer function on output. The phase of the output would be undistorted for frequency components at or below the breakpoint of the filter (see Figure 8).

When the breakpoint frequency of the filter is changed, the scale of Figure 7 would have to be changed accordingly. The magnitude response can be thought of in terms of percent of the breakpoint frequency and percent of modulus transfer. Between zero and one hundred percent of the breakpoint frequency the magnitude response is 100 ±1 percent. Past the breakpoint the response rolls off sharply to about 2 percent at 155 percent of the breakpoint frequency, and nearly zero at 180 percent of the breakpoint.

The performance of this filter begins to deteriorate as the breakpoint is set below 1/80th of the original sample rate. Thus, data digitized with a sample rate of 4096 should not be filtered with a breakpoint much less than 50 Hz. This restriction applies only to the digital filter used in the File Creation Program.

When the user selects a sample rate reduction factor, aliasing may be prevented by also selecting a filter breakpoint sufficiently low so that noise components with frequencies at or above the new Nyquist point are adequately attentuated. The filtering capability can also be used to create smooth data which can improve the readability of plots. However, frequency components important to the data must not be attenuated significantly.

Those users who specify filtering for their data should be aware of the problem of edge effects or filter initialization. When generating filtered values near an edge of the specified output record, the filter must include in its calculations input data points corresponding to times outside of this edge. If such data are not present, erroneous filtered output can be generated for those values near the edge. If data outside the output record boundary is present on the data tape (if an offset is specified for example), that data will be accessed

by the program for the filter to use. If such data are not available, a warning message is generated in the printout from the File Creation Program.

Operation of the Processing Program can be optimized by reducing the sampling rate for the True Airspeed, OAT and Boom System static pressure and filtering with as low a breakpoint as possible. However, the rotor azimuth data should not be filtered nor should its sample rate be reduced.

3.2.5 Time Alignment

For the OLS application of this system, an option is provided in the File Creation Program to correct for various time misalignments between the data from different sensors. Specification of the ALIGN command in the File Creation Program input will cause the program to read a tape containing a correction offset for each item code/counter pair. The correction offset will then cause the program to discard a certain number of points at the start of the data stream in the same fashion as an OFFSET. If an ABSOLUTE offset is specified for a counter, all time alignment offsets are ignored.

> **WARNING**
>
> When OLS data are being processed, the ALIGN option should always be selected unless the user has specific information that alignment offsets are invalid.

3.3 QUESTION AND ANSWER PROGRAM TO CREATE USER INPUT DATA SETS

This special program is provided for users to develop complete input sequences for the File Creation Program in an interactive mode. Questions are asked to prompt input for every possible specification. The output of the program is a data set which can then be specified as the user input for the File Creation Program. This program will assure the user that his input for the File Creation Program is correct in syntax and that all possible specifications have been considered. This program cannot assure that the content of the generated input stream is correct. For example, this program cannot assure that a specified item code actually corresponds to the data the user wants.

Upon entry to this program, the user is guided step by step on a question-and-answer basis through all possible inputs. Although knowledge of the syntax for the File Creation Program is unnecessary, the user must be familiar with the effect of the various inputs (for example the user must know that OFFSETs apply to all the item codes for a given counter). All of the necessary information regarding both the data to be transferred and the Master File partition must have been gathered before the user executes this program.

3.4 FILE MAINTENANCE

When the Master File is accessed by many users, there should be a designated data base monitor. This monitor must perform three functions. First the information on the Master File must be protected against some catastrophic loss of data (e.g., a disc hardware failure). Second, the storage requirements of the users must be coordinated. Third, to assure the availability of adequate storage space, the data present on the Master File must be scanned to assure that unused partitions are deleted. In the event that there were only a very few users in good communication with each other, these users might share the monitor function.

The File Maintenance Program is provided for use by the monitor in fulfilling the requirements listed above. The functions provided by this program are listed in Table 3 along with the commands which access those functions. One command and one qualifier is allowed per line of input to the program and the program executes the commands in the sequence in which they are read.

The partition map is the same as the map which can be written by the File Creation Program after data transfer is complete. The Master File MAP command lists each partition name along with offset to the first record, size in records, user, creation date, last access, and password. The Master File password (SUPERWORD) must have been entered before a Master File map can be generated.

The partition SAVE command copies a partition to digital tape for backup or archival use of the data. A Master File SAVE command copies the entire Master File to digital tape. The user must take care in using either SAVE that the partition records will fit on one digital tape. Currently, one Master File record contains 1024 bytes and the user must determine from this figure, the number of records in the partition in Master File, the JCL, and the tape size whether the data will fit. Only one SAVE (partition or Master File) can be performed per run of the File Maintenance Program.

TABLE 3. FUNCTIONS AND COMMANDS FOR THE FILE MAINTENANCE PROGRAM

FUNCTION	COMMAND
Map of a partition content	MAP (PARTITION NAME)
Map of partitions in the Master File	MAP ALL
Save on digital tape of a partition	SAVE (PARTITION NAME)
Save on digital tape of the Master File	SAVE ALL
Copy a partition from digital tape to the Master File	RESTORE (PARTITION NAME)
Copy a Master File image (SAVE) from digital tape to empty Master File	RESTORE ALL
Delete a partition from the Master File	DELETE (PARTITION NAME)
Enter a partition password	PASS (PASSWORD)
Enter password for the Master File	SUPER (SUPERWORD)
End processing	END

The partition RESTORE command copies a SAVE'd partition from digital tape back to the Master File. The partition name specified with RESTORE cannot already be present on the Master File (see DELETE below) and there must be sufficient free storage space to accommodate the partition size.

The Master File RESTORE ALL copies a SAVE'd Master File image from digital tape back to the Master File disc area. The Master File disc area to be written on must be initialized, contain no partitions, and be sufficiently large to hold all the Master File records on the tape.

The partition DELETE command removes a partition name from the Master File. Either the SUPERWORD or the partition PASSWORD must have been specified to DELETE a partition.

The SUPERWORD is a special password provided for the use of the Master File monitor. This password is defined during the Master File initialization run. The function of the SUPERWORD is to allow the monitor to DELETE partitions without knowing the individual password for each partition.

The File Maintenance Program can be run either interactively or in batch. The user should be aware that in interactive mode the MAPs will be printed on the terminal being used. No other program which accesses the Master File may run while the File Maintenance Program is executing.

4. PROCESSING PROGRAM - GENERAL ASPECTS OF USER INPUT

4.1 GENERAL USER INPUT RULES

User instructions to the Processing Program follow the same general free-field rules as the input to the File Creation Program. However, the structure and meaning of the commands are different. Commands are typed in as sequences of entries. Two entries are separated by one or more blanks, a comma, or a slash. The slash is a special separator which is explained later. An entry is a sequence of characters which does not include a separator. There are three kinds of entries: numbers, strings, and nulls.

A number begins with a digit, decimal point, plus sign, or minus sign. Subsequent characters can include digits and/or a decimal point. Only one decimal point is allowed in a numeric entry. Exponential (power of ten) numbers are not allowed. Decimal points are not required for floating numbers if no fractional part is included. Following are some examples of valid and invalid numeric input:

Valid	Invalid
1024	.126.4
1024	10.5E-6
+10.2458	128A5
.1024	-10.4-3

A string entry begins with any character other than a separator, digit, plus sign, minus sign, or decimal point. Subsequent characters in a string can be anything except separators. Following are examples of valid and invalid string entries:

Valid	Invalid	Reason Invalid
P823	8PAT	Starts as number
CONTOUR	IN,DY	Contains separator
AZY+*!		

A null entry is indicated by two commas in sequence. In some cases, null entries are used to imply default entries (see Section 4.4).

A line consists of a sequence of entries contained in character positions 1 through 55. The particular character position where an entry begins is not important in evaluating a number

or identifying a string. The sequence of entries is vital. An entry cannot extend from the end of one line to the beginning of the next.

When string entries are to be identified for meaning, the characters are matched with a list of model character sequences. The characters are matched character by character for the first four characters. If fewer than four characters are input, then the match can still be made so long as it is unambiguous. Thus, the model entry for HARMONIC is HARM. However, HAR is an adequate match for HARM since it is unambiguous. Some more examples are:

DER = DERI = DERIVE
DIS = DISP = DISPLAY
ED = EDIT
EXE = EXEC = EXECUTE

Extra characters in an entry up to the fourth character <u>must</u> match. Thus,

DERX ≠ DERI
EDZZ ≠ EDIT

4.2 <u>COMMAND STEPS AND SUBSTEPS</u>

Commands will be input and executed in <u>steps</u>. A step is a self-sufficient action by the program in that the code can proceed without intermediate input from the user (exceptions to the rule are CHANGE mode of EDIT, printout control, MENU listing control, and cursor control for plots) and some tangible result is produced - either an output, an intermediate storage of data, or a change in state of the program run. No subset of a complete step will produce tangible results from the computer. In fact, a step which is incompletely entered can be erased at any time with a 'CANCEL' control command and the computer is returned to the status held before starting the step.

A step will include one to four <u>substeps</u> which fall into four categories: SPECIFICATION, ACTION, INPUT, DISPOSITION. A SPECIFICATION is always required. The other substeps may or may not be needed depending on the nature of the SPECIFICATION. Table 4 is a list of the specifications, which are all one entry, along with associated required substeps:

TABLE 4. REQUIRED SUBSTEPS FOR EACH SPECIFICATION

SPECIFICATION	ACTION	INPUT	DISPOSITION
ANALYZE	X	X	X
DERIVE	X	X	X
DISPLAY		X	X
EDIT		X	
BUILD		X	
SAVE			
NOEDIT			
EXECUTE		X	
MENU		X	
TERMINATE			
COMMENT		X	

If a substep is not required for a particular specification, the input routines will assume it is not present. Thus, substep input following the specification substep DISPLAY will be assumed to belong to the 'INPUT' substep rather than an unnecessary 'ACTION' substep.

The 'SPECIFICATION' substep indicates the sort of process to be performed in the step. Sometimes the specification substep entry does not require amplification in the form of additional substeps (e.g., 'TERMINATE'). Usually, additional information is required to define the step so that one or more of the other substeps are required.

The 'ACTION' substep indicates something to be done to the data after it is input. Examples of actions would be a filtering process or derivation of C_n. Generally, the 'ACTION' substep is required to define the specification substeps ANALYZE and DERIVE more precisely.

The 'INPUT' substep specifies where the data for processing and display is obtained. It can specify the number of dimensions of the input and independent variables to be used. Specialized entries are required for such specifications as 'EDIT' and 'COMMENT'.

The 'DISPOSITION' substep describes the form of the output of the step (e.g., contour plot, save on scratch disc file, etc.).

Substeps are separated from each other by slashes. Thus, more than one substep can occur on a line, or a substep could take more than one line to input. Conceivably, an entire complex step could be entered on one line.

4.3 DEFAULTS

The use of default entries can be of value to the user for two reasons. First, a default is a quicker entry, saving the user time. Second, a default requires fewer character inputs, which reduces the possibility of error during input. In using defaults, however, the user bears the responsibility of knowing what the defaults are and how the default specifications and structure work.

Default values and options are specified through more than one procedure. Some parameters have defaults that are permanently specified in the program. Other parameters retain the value last specified during the run. Sometimes a parameter is set to an initial value at the start of a run or session and then changed as nondefault values are specified. For other parameters, no default exists until a value has been specified by the user once; an attempt to default to the uninitialized value produces an error message. (For example, there is no default counter as a user session begins. Once a counter is specified, that counter is the default until a new one is specified.)

Both single entries and groups of entries can be defaulted. The simplest default for an entry occurs when that entry is the last of a substep. The entry is simply omitted, and the slash marking the substep end is inserted immediately. For example, instead of:

```
HARMONIC, 12 /
```

the user could leave off the last entry since "12" is the default number of harmonics:

```
HARMONIC /
```

A group of entries at the end of a substep can be defaulted in the same fashion. Thus:

```
CP, 264, CALC, CALC /
```

could be entered

```
CP /
```

If an entry to be defaulted occurs before an entry in the same substep which must be entered, then the default can be specified by the presence of two comma separators in sequence. For example,

MCQ,5,,80,13/

would cause the third entry, rotor radius, to default to a previously entered value. More than one default in sequence can be specified by more than two commas. For example,

MCT,5,,,80,13/

would cause the third entry, ship gross weight, and the fourth entry, rotor radius, to default to previously entered values.

4.4 ERROR HANDLING

User-input errors detected by the program fall into two major categories, invalid line and invalid entry, depending upon the effect of the error on the input sequence.

Invalid line errors consist of two types. The first is numeric entries which cannot be successfully interpreted as numbers by the free-field input routine. Such errors include embedded illegal characters in an entry that begins with a numeric entry character (+, -, ., 0-9), and numeric entries with multiple decimal points. The second type of invalid line has characters typed beyond column 55.

An invalid line error will cause the program to output a line containing only one character, a '+', pointing to the position where the error was detected. In the interactive mode, the user must then retype the entire line correctly and proceed as before. In batch mode, the program terminates without execution after scanning the remaining lines of input for error.

An invalid entry error is essentially any other error which can be detected in the user interface. When such an error is detected in an entry, that entry is invalidated and all subsequent entries on the same line from the same or subsequent substeps are invalidated. Previous entries on the same line remain valid. The program prints a message indicating the erroneous entry, and converts to HELP mode for the remainder of the substep (see Section 4.5).

Not all errors can be detected in the user interface part of the program. A command step may have proper syntax and structure but be impossible to execute. For example, the user might request a True Airspeed derivation although data from the airspeed sensor is not available for processing. When errors such as these are encountered in processing, an error notification and number is output. Frequently, a diagnostic is output. Occasionally, an error is only identified by an error number. These numbers are identified in Appendix A.

4.5 CONTROL COMMANDS

Certain string entries are recognizable throughout the specification of a step. These control commands do not constitute values for the entry positions but instead require some instant action on the part of the input routines. Subsequent entries on the same line are ignored by the input routines. The control commands are HELP, LIST, and CANCEL.

4.5.1 HELP Command

HELP, or the alias '?', will cause the program to prompt the user for proper input. If HELP or '?' is substituted for any entry of a substep, including the first, the user will be instructed as to the meaning of each remaining entry and, where possible, which allowed options are available for each entry of that substep. Sometimes the user will be asked for one entry at a time in the HELP mode, since each entry can affect the meaning of subsequent entries in the substep. On other occasions, entire sequences of entries will be requested from the user.

A specific example will now be considered. An exchange between the computer and the user making maximum use of HELP is shown and then explained line by line. User entries are underlined while computer outputs are not. Line numbers are included for reference only and would not appear in an actual user/computer interchange.

1. NEW STEP.
2. <u>HELP</u>
3. SPECIFICATION SUBSTEP
4. ENTER:- (1)OPTIONS: ANALYZE,DERIVE,DISPLAY,EDIT,NOEDIT
5. ,EXECUTE, MENU, TERMINATE, START, SAVE, COMMENT

6. DISP

7. SPECIFICATION SUBSTEP COMPLETE /

8. INPUT SUBSTEP

9. ?

10. INPUT SUBSTEP

11. ENTER:- (1)INPUT SOURCE: SCF1,SCF2,GROUP,(ITEM CODE)

12. A938

13. ENTER:- (1)(COUNTER); (2)TIME(SEC); (3)DURATION(SEC)

14. 676 0 1

15. INPUT SUBSTEP COMPLETE /

16. DISPOSITION SUBSTEP

17. ?

18. DISPOSITION SUBSTEP

19. ENTER:-(1)OUTPUT OPTIONS: PLOT,MPLOT,APLOT,PRINT,CONT

20. OUR,SURFACE,KEEP,ADD

21. PLOT

22. ENTER:-(1)1ST AXIS OPTIONS: TIME,FREQ,HARM,ROW,COLUMN

23. ,MRAZ,RPM,TAS,IMPLIED; (2)CURSOR OR CLOSE;

24. (3)INTERVAL Y: AUTO,LOG,(INTERVAL)

25. TIME CLOSE AUTO

26. ENTER:- (1)BOTTOM Y: AUTO,(Y); (2)INTERVAL X:

27. AUTO,LOG, (INTERVAL)

28. AUTO AUTO

29. ENTER:- (1)MINIMUM X: AUTO,(X)

30. AUTO

31. DISPOSITION SUBSTEP COMPLETE

32. WAITING FOR '/' TO EXECUTE STEP

33. /

34. EXECUTING

In line (1), the program informs the user that the previous step has completed or aborted and a new step is beginning. Line (2) informs the computer that the user does not know what to do. The computer responds with line (3), which informs the user that input will be for the specification substep, and with lines (4) and (5), which list the entry options. The user selects DISPLAY on line (6). On line (7) the computer indicates that the specification substep is complete. A slash is included to substitute for the slash the user would have entered if HELP were not active.

Since the program exited the specification substep in the HELP mode, the user is told the new substep name in line (8) and the HELP mode is terminated. The user still cannot remember what to do, however, and so requests more HELP with line (9) using the '?' alias. In line (10), the computer reinforms the user that the current substep is 'INPUT' and lists possible first entries on line (11). The user responds with an item code entry in line (12).

In line (13), three different entries are prompted at the same time. The computer requests the counter, the time offset from the start of the time history, and the time duration of data to use. In line (14), the user responds by entering counter 676, time offset zero, and record length one second. Since all entries for the INPUT substep are now specified, the user is advised that the substep is complete in line (15) and a slash is supplied by the computer. Then, in line (16), the computer informs the user that the new substep is DISPOSITION.

The user requests additional HELP in line (17), and the computer responds in line (18) that the current substep is DISPOSITION. In lines (19) and (20), the output options are listed for the first entry of the DISPOSITION substep. The user selects a simple X-Y plot in line (21). In lines (22) through (24), the user is requested to define the independent axis variable, the status of the Tektronix crosshair cursor following generation of the plot, and the scaling interval for the dependent plot axis. In line (25), the user selects TIME as

the independent variable axis, no cursor activation, and automatic selection of the Y (or dependent variable) axis scale interval. The program then requests the minimum 'Y' axis value to depict and the scaling interval for the 'X' (or independent variable) axis in lines (26) and (27). Automatic scaling is chosen by the user for both cases in line (28). The program then requests the minimum 'X' value to depict in line (29) and the user again selects automatic scaling in line (30).

The DISPOSITION substep is now complete and the computer announces this fact in line (31) without inserting a slash to terminate the substep. In line (32), the computer informs the user that a slash must be entered to execute the step. The user provides the slash in line (33), and the computer responds in line (34) that the step is being executed.

These 34 lines used to define one command step may appear unduly time-consuming. Notice, however, that if the HELP mode is not invoked, the above 34-line transaction may be accomplished by the following three lines:

1. NEW STEP.

2. DISP/A938 676 0 1/ PLOT TIME CLOS AUTO AUTO AUTO AUTO/

3. EXECUTING

By using defaults, line (2) above may be further reduced to:

(2) DISP/A938 676 0 1/PLOT/

Some important features of the HELP format can be observed in the example given. The HELP messages are indented at least one column position. In fact all program output is indented at least one column position. User input can begin in column one. In the interactive or interactive-graphics mode, this feature should distinguish user inputs and program messages.

A HELP message specifically requesting one or more entries will always begin with 'ENTER:-'. More than one entry can be prompted in one message and the individual entry prompting messages are identified with a number enclosed in parentheses. This number refers to the entry position which the program is currently waiting for rather than the entry number in the substep. Entry options are separated by commas while separate entry prompting messages are separated by semicolons.

Specific keyword string options are not enclosed in parentheses. Numeric value options or nonkeyword string options are signified by either enclosing a descriptor in parentheses or enclosing the corresponding units in parentheses. For example, in line (27) above, 'AUTO' is selected with the literal string 'AUTO' while in line (12), '(ITEM CODE)' is selected by entering the item code 'A938'.

The HELP command can be used in conjunction with defaults. As before, commas can be used in sequence to specify default entries for a substep. For example, line (24) above could be entered:

(24) <u>TIME,, AUTO</u>

since CLOSE is the default response for the Tektronix cursor activation status. The slash can also be used to specify defaults for the last entries in a substep. Line (24) could be entered:

(24) <u>TIME/</u>

(25) EXECUTING

where all the remaining DISPOSITION substep entries are set to defaults and execution of the step begins immediately.

HELP mode is immediately terminated when the slash is entered. The user then has the option to specify entries for subsequent substeps on the same line. For example, line (14) could be entered:

(14) <u>676 0 1/PLOT/</u>

(15) EXECUTING

In fact the user always has the option to specify additional entries which follow those actually prompted by a HELP message.

4.5.2 <u>LIST Command</u>

If HELP or defaults have been used in generating a command step, the user may become uncertain about the exact entries used. By specifying LIST, the user can obtain a complete listing of the entries already specified in the step input, including any defaults. LIST can be entered at any time prior to entry of the slash terminating the final substep. The listing will include all entries specified for the step up to

the entry just prior to the position occupied by the LIST command. However, if LIST is entered before any entries are made for a step, the previous step will be listed. This feature can be of considerable use if a step execution has gone wrong and the user wants to find out why.

After a LIST is completed, the program will expect input entries beginning with the entry position for which LIST was substituted. If the program was not in the HELP mode when the user entered the LIST command, HELP will not be active when the LIST is complete. If HELP was active when the LIST command was entered, then HELP will still be active when the LIST is complete and the user will be reprompted for entries beginning with the entry the LIST command replaced. If the user entered LIST after typing all the entries for the final substep but before the final slash, the program will still be waiting for that slash after the LIST is complete.

4.5.3 CANCEL Command

By entering CANCEL, all input entered for the step to that point is canceled and the program returns to the beginning of a new step. CANCEL at the beginning of a step is ignored.

4.6 COMMAND SEQUENCING (Edit)

The Edit or Command Sequencing function allows the user to create, modify, execute, or delete sequences of command steps. These sequences are stored on a permanent disc file so that sequences created in one run of the Processing Program can be executed in other runs of this program. In these separate runs the operating mode (i.e., Batch, Interactive, Interactive Graphics) may be different. There are three motivations for using the Edit or Command Sequence capability:

- **The user may want to enter instructions interactively using the error checking process, but without execution of steps, and then execute the identical instructions in batch mode.**

- **The user may want to enter instructions interactively and see the results immediately and then get incremental plotter and line printer versions of printout and plots from batch without retyping the instructions.**

- **The user may have long sequences of instructions to enter which are generally the same for all given executions but which involve a small number of changes.**

Commonly, the user may have a combination of these motivations in using the Edit process.

This process involves creation, modification, and deletion of blocks of instructions. A block will consist of a sequence of one or more steps of user instructions and is limited to no more than 112 lines of 55 characters, regardless of the number of command steps included.

A block of instructions can be used by entering EXECUTE as the specification substep and the block name in the input substep. The instruction lines are extracted from disc in sequence just as though the user is entering them. When the program encounters a NOEDIT specification substep, it exits the EDIT block execution mode and reenters the direct mode. If an error is encountered while in block execution mode, the program returns an error message and exits this mode.

Blocks of instructions can be created by using the EDIT specification with NEW entered in the INPUT substep or by using the BUILD specification. If EDIT with NEW is specified, then the program enters a mode such that commands are read with normal error checking. However, when the final slash is entered requesting execution of the step, then all the lines comprising the step are copied to the Edit block on disc and the step is not executed. Note that error checking is done on input syntax or unreasonable or unrecognizable entries but not on errors which would be detected in executing the step. Thus, if the user requests a PLOT of processed data which is not stored on the scratch disc file specified for input, the error would be undetected. The user is free to use HELP, CANCEL and LIST while in the EDIT mode as control commands will not appear in the command chain created.

Exit from this mode is accomplished with a NOEDIT specification substep. At that point, a NOEDIT specification substep is written for the last line of the Edit block. The block is available with the assigned name in the Edit file.

The second way to create Edit blocks is by using the BUILD specification. An Edit block is started and the program returns to direct mode. The program actually executes each step when the final slash is entered. The user can then record the lines of instruction comprising the previously executed step by specifying SAVE as the next step. An intervening LIST control command will not interfere with the SAVE. The user completes the Edit block by entering NOEDIT as in the EDIT/NEW method.

The BUILD and SAVE specifications allow the interactive graphics user to actually try out the execution of steps and save only the commands which produce desired output. Data can be plotted on the Tektronix, scanned for correctness, and then the same instructions can be executed in batch to produce Calcomp or DP-1 plots. Of course, the user must also SAVE intermediate steps which process and store data. SAVE'd processing instructions which later appear superfluous or incorrect can be removed with the CHANGE mode of EDIT.

Creation of Edit blocks with the EDIT/NEW mode provides advantages different from those provided by the BUILD mode. First, commands which create plots can be entered on a non-graphics terminal. Second, commands are entered with syntax checking but without execution so that considerable time can be saved in generating a command sequence block.

Two other modes of EDIT are available: CHANGE and DELETE. DELETE is quite simple. The Edit block name is removed so that the space and the name are available for later use. The CHANGE mode is more complicated and is not envisioned for use by brand new users. CHANGE allows the user to alter an existing Edit block. The user must be cautious in the change mode since line modifications are checked only for line errors (see Section 4.1) and syntax checking is not performed on the resultant instruction sequence until the Edit block is actually executed. Paragraph 5.7.1 provides additional information on the CHANGE mode.

4.7 'INFO' FILES

During execution, the program will frequently need to access certain item codes for derivations (e.g., the rotor azimuth item code). The program will also need to access groups of item codes which measure like parameters along with geometric positions and matrix (row, column) organization for these item codes. If the user was required to enter all this information for every command step requiring it, the process would prove time-consuming, tedious, and prone to error.

The 'Info' file is provided to minimize such repetitive entries. This file consists of two parts. The first part is a short list of keywords with corresponding critical item codes such as rotor azimuth and boom system airspeed. The second part consists of several groups of item codes of like kind. Each group has a four-character name and includes labeling information, row (chord) and column (radius) position, and sensor grouping information along with the corresponding item codes for the row-column intersections. A group can have

multiple rows and columns (a two-dimensional group) or a single row and several column positions. Groups can specify both double-row elements or only a single double-row element.

The file format is sequential and consists of a number of 70-column lines which can be maintained as such in 70-column card image form. For the OLS application, two specific Info files are provided. However, a user may create special Info files for different applications containing different item codes and different geometric information. Rules for the structure of an Info file are contained in Section 5.6.

When specifying input data with the Info file, the user simply references the appropriate group by name. Figure 25 lists the groups in the format specified in Section 5.8. Entry options are then provided for the user to select a single element or all elements in both the row and column directions. The user also has the option to select BOTH double-row elements or either the TOP or BOTTOM double-row element. For example, in the OLS application using the absolute pressure sensor group, S2PP, the user could select ALL for the row elements (chord positions), '3' for the column elements (75 percent radius position), and BOTTOM for the lower surface sensors.

Two separate Info Files are provided for the OLS blade application, one Info File for each blade. The different Info Files supply appropriate corrective factors to properly align azimuth for the different blades. Thus, the proper Info File should be used as indicated at each installation for red blade sensor or white blade sensor applications.

5. PROCESSING PROGRAM-SPECIFIC USER INSTRUCTIONS

5.1 SPECIFICATION SUBSTEP COMMANDS

The Specification substep has a single-entry position. There are 11 allowed keywords for this entry which were listed in Table 4. The function of each of these keywords is given here.

ANALYZE and DERIVE each specify processing to be performed upon data. Data are accessed, processed, and output with either of these specifications. The actual process to be performed is identified in the Action substep (Section 5.2). An ANALYZE function performs some process on a general class of data (e.g., time histories). A DERIVE function performs a process on particular measured and/or derived parameters to produce a specific derived parameter output. DISPLAY has the same capabilities as ANALYZE and DERIVE for input and output. However, DISPLAY performs no processing function.

EDIT specifies one of three Command Sequencing functions-NEW, CHANGE, or DELETE-identified in the Input substep. The meaning of these functions was covered in Section 4.6. Specific instructions for use of the CHANGE mode of EDIT to modify sequences of Command Steps are given in Section 5.7.1. The BUILD command was also covered in Section 4.6 along with SAVE, EXECUTE, and NOEDIT.

The COMMENT specification provides a user capability to enter additional labeling on plots and printouts beyond the labels normally provided by the program. The comment itself is entered as a special string in the Input substep which may contain blanks and commas but not slashes. The comment is limited to 55 characters from the first nonblank character to the last nonblank character. The comment entry may not contain sequences of characters which create an invalid line error. For example:

COMM/THIS IS THE 2ND FILTERED CASE/

would be unacceptable because '2ND' creates an invalid line error. The comment may appear in a line separate from the Specification substep. Thus:

COMMENT/

THIS IS ONE VERY LONG COMMENT USING FIFTY-FIVE COLUMNS/

is acceptable.

The MENU specification allows the user to request displays helpful in determining how to process the data. One such MENU is a list of the Groups present on the 'Info' file. All the possible MENU's are listed with explanation in Section 5.3.

The TERMINATE specification causes the program to halt. A message is printed:

RUN TERMINATES

and the program is exited.

5.2 ACTION SUBSTEP COMMANDS

The ACTION substep entries define the process requested by an ANALYZE or DERIVE specification substep. Many of the initial ACTION substep entries are followed by additional qualifying entries. Some of these qualifiers have default values. Each initial ACTION substep entry will be explained here along with qualifiers and default for qualifiers. Parentheses in the listing only signify that a number or string should be substituted in the corresponding entry location. The parentheses do not appear in the actual command step.

5.2.1 ANALYZE Commands

HARMONIC, (Number of Harmonics), (Harmonic Number)

This command generates a Harmonic Analysis for an integral number of rotor cycles. The number of cycles is defined in the Input substep. The second entry in the ACTION substep gives the number of Harmonics to extract (the zeroth Harmonic is ignored). The third entry in this substep (Harmonic Number) gives the specific Harmonic number to calculate if the second entry is one. Harmonic analysis produces two double-row elements (TOP = amplitude, BOTTOM = phase) for output. The rotor cycles can be defined by the main or tail rotor as selected at the beginning of the program run (see Section 5.6).

The default for the third entry is one and the default for the second entry is twelve. No more than 24 Harmonics may be specified. Examples of use of this substep are:

ANALYZE/HARMONIC 6/...

or

ANAL/HARM, 1, 4/...

FILTER, (Upper Break Freq), (Lower Break Freq), (Number Poles)
This command causes measured data or processed data to be digitally filtered. The pass band is the frequency interval between the frequencies: (Upper Break Freq) and (Lower Break Freq) in Hz. If (Lower Break Freq) is set to zero, the filter becomes a low pass filter. The rate of roll-off (see Paragraph 6.1.2) outside the pass band is set by (Number Poles). Increasing the number of poles in the filter increases the roll-off rate. Normally three or four poles are adequate. The allowed range for (Upper Break Freq) is between 0.1 and 100,000 Hz with no default. The allowed range for (Lower Break Freq) is between 0.0 Hz and (Upper Break Freq) with a default of 0.0 Hz. The allowed (Number Poles) is two through seven with a default of four. Examples of this substep command are:

ANALYZE/FILTERS, 50, 0, 3/...

ANAL/FILT 100/...

SPECTRUM, (Maximum Frequency), (Window)
This command generates an Amplitude Spectrum of a measured or processed time history. Frequency components are displayed between the reciprocal of half the record length and (Maximum Frequency) in Hz. (Maximum Frequency) must be between 0.0 and 100,000 Hz. If (Maximum Frequency) is set to 0.0 or is greater than the Nyquist Point for data, the corresponding limit is set to the Nyquist Point during the processing phase. (Window) specifies a data window function to be applied to the time history before the spectrum is calculated (see Section 6.1). The allowed windows are COSINE TAPER, HANNING and RECTANGLE. The default (Maximum Frequency) is 0.0, which is reset to the Nyquist Point in processing. The default for (window) is COSI (cosine taper). Examples:

ANALYZE/SPECTRUM 200 RECT/...

ANALYZE/SPEC/...

DAMPING, (Damping Frequency)
This process calculates an estimate of the percent of critical damping found in a time history for the specified frequency, (Damping Frequency). Damping analysis process produces a single output value - percent of critical damping - for each individual time history that is input. Thus, a DAMPING analysis of a single time

history cannot be plotted (the value can be printed). DAMPING analysis must be performed on a group of time histories to generate sufficient points for a plot. The range of (DAMPING FREQUENCY) is 0.0 through 100,000, but a frequency higher than the Nyquist Frequency for the data will produce a processing error (number 771). There is no default for (Damping Frequency). An example:

ANAL/DAMPING 112/...

AVERAGE

This process creates one representative time history, one rotor cycle in length, by averaging together several contiguous rotor cycles of data from one sensor. When the process is performed, the data rate for the output is arbitrarily set so that there are 256 output samples in the single cycle of data produced. Main or tail rotor azimuth can be used for specifying these data cycles as specified by the rotor selected at the beginning of the program run (see Section 5.6). An example:

ANAL/AVERAGE/...

MMAX

This process generates min/max data from time history data. For every rotor cycle of each input time history, the minimum (min) and maximum (max) values are found. From these values, mean and oscillatory values are calculated (see Section 2.4 or 6.1). The oscillatory is output as the TOP double-row element and the mean as the BOTTOM. Rotor cycles are defined by the main or tail rotor as specified at the start of the program run (see Section 5.6). The output is a time history with the first value located in time at the midpoint of the first cycle and having a data interval one rotor cycle in length. An example:

ANAL/MMAX/...

5.2.2 <u>DERIVE Commands</u>

TAS, (OAT), (Static Pressure), (Correction Slope), (Correction Intercept)

This process derives Vehicle True Airspeed from measured indicated airspeed and the parameters specified by the qualifiers above according to the method described in Section 6.2. Entry number two, Outside Air Temperature (OAT in degrees Celsius) and entry number three, Static Pressure (PSIA), can be specified either with a constant

numerical value or the word CALCULATE, which means that smoothed, measured data will be used for these parameters (see Section 2.7). Both of these entries default to 'CALC' initially and then to the previous setting. Static Pressure and OAT are common to many derived parameter calculations, and the previous setting default means the setting specified in the most recent such derivation. Entries four and five correspond to slope and intercept values for the conversion from True Indicated Airspeed to Calibrated Airspeed (see Section 6.2). The initial defaults for these numbers are 1.0 and 0.0, respectively. When new values are specified for these two entries, these new values become the defaults until reset or the end of the program run. Allowed values for the slope are .5 to 2.0 and allowed intercept values are -40.0 to +40.0.

TAS is an 'attached parameter' (see Section 2.7) and has the following attached parameter characteristics: (1) Once calculated for a given counter and time span, the parameter is not recalculated until a new counter or time span is specified, (2) the parameter values are smoothed a great deal, and (3) the parameter output data stream contains one value for each rotor cycle.

Here are some examples of TAS commands:

DERIVE/TAS CALC 14.23/...

DERIVE/TAS CALC CALC 1.014 -0.013/...

DERI/TAS/...

MRPM

This process calculates the rotor speed in units of RPM. The main or tail rotor data will be displayed based on the 'rotor mode' defined at the beginning of the program run (see Section 5.6). An example:

DERI/MRPM/...

MSHP

This process calculates the Mast Horsepower as described in Sections 2.4 and 6.2. The shaft torque item code as well as the rotor azimuth item code must be present for this calculation. The main or tail rotor will be selected as above. Example:

DERI/MSHP/...

CP, (Rotor Radius), (OAT), (Static Pressure)

This program calculates Blade Static Pressure Coefficient from Blade Absolute Pressure Sensor data along with all of the attached parameters (Section 2.7), rotor radius, and the radial station on the blade expressed as a fraction of the total radius. See Section 6.2 for a description of the process. The second entry is rotor radius in inches from hub to tip. There is no initial default for this value but, once set, the most recently entered value becomes the default. The most recent rotor radius entry in any of several derivations will be used as the default.

The third and fourth entries, OAT and Static Pressure, are the same as the OAT and Static Pressure entries for the True Airspeed (TAS) derivations. Examples:

DERI/CP, 264, CALC, 14.23/...

DERI/CP,,,14.50/...

DERI/CP/...

CN

The Blade Normal Force Coefficient is calculated from Blade Static Pressure Coefficient, C_p, data integrated around all the chord positions for a radial station. The only possible input source for the C_n process is a scratch file (SCF1 or SCF2) containing C_p data. The command step which generated that C_p data must have used the Info file to generate values from a sufficient number of sensor positions around the chord for a C_n integration to be successful. Example:

DERI/CN/...

CC

The Blade Chordwise Force Coefficient is calculated from C_p data in similar fashion to C_N. Example:

DERI/CC/...

CM

The Quarter Chord Blade Pitching Moment Coefficient is calculated from C_p data in similar fashion to C_N. Example:

DERI/CM/...

MCT, (OAT), (Static Pressure), (Vehicle Weight), (Rotor Radius)

This process calculates the Thrust Coefficient as described in Section 6.2. Entries two and three, OAT and Static Pressure, are as described under the True Airspeed (TAS) derivation. The fourth entry is Gross Vehicle Weight (GVW) in pounds. There is no initial default for GVW but, once set, the most recently entered value becomes the default. The fifth entry, Rotor Radius, is described under the C_p derivation. Examples:

DERI/MCT 30 14.4 8300 264/...

DERI/MCT,,, 9000/...

DERI/MCT/...

MCQ, (OAT), (Static Pressure), (Rotor Radius)

The Torque Coefficient, C_Q, is calculated as described in Section 6.2. The second and third entries, OAT and Static Pressure, are described under the TAS derivation. The fourth entry, Rotor Radius, is described under the C_p derivation. Examples:

DERI/MCQ, CALC, CALC, 264/...

DERI/ MCQ, 32.5,, 264/...

DERI/MCQ/...

MFLO, (OAT), (Static Pressure), (Angle)

The Blade Local Flow Magnitude (in ft/sec) is calculated as described in Section 6.2. When this command is selected, both the Flow Magnitude and Direction (in degrees) are calculated with the Magnitude output as the TOP double-row element and the Direction as the BOTTOM double-row element. When the output is printed, both double-row elements are shown, but plotted output can display only the TOP double-row element (Flow Magnitude in this case).

Entries two and three, OAT and Static Pressure, are described under the TAS derivation. The fourth entry, (Angle), is the angle in degrees formed by the inboard pointing Boundary Layer Button (BLB) sensor and the chordline. The initial default for (Angle) is 45 degrees. This number is replaced by any nondefault entry. Examples:

DERI/MFLO, CALC, 14.3, 43.5/...

DERI/MFLO/...

DFLO, (OAT), (Static Pressure), (Angle)
The Blade Local Flow Direction command is identical to the 'MFLO' command except that the Flow Direction becomes the TOP output double-row element for plotting output and the Flow Magnitude becomes the BOTTOM. On output, positive angles mean outboard to inboard flow and negative angles mean inboard to outboard. Angles are measured in degrees from the chordline. Examples:

DERI/DFLO CALC CALC 46.1/

BLDISP, (Harmonic Number)
Local Blade Displacement, in inches, is calculated from accelerometers on the blade which are oriented in either the chordwise or beamwise direction. The displacement is calculated for a single harmonic of the rotor cycle according to the process described in Section 6.2. Section 6.2 also documents deficiencies in this derivation method. The second entry, (Harmonic Number), is the single rotor cycle harmonic number and has allowed values of 1 through 24. The default harmonic number is 1. One complete rotor cycle of 256 displacement values is always produced by this derivation. Examples:

DERI/BLDIS 2/...

DERI/BLDIS/...

SLOPE, (Rotor Radius)
Local Blade Slope is calculated from Local Blade Displacement derivations for multiple radial positions on the blade. Thus, the input source for this process must be a scratch file (SCF1 or SCF2). The output of the process is unitless (inches/inch). The second entry, (rotor radius), is as described under the C_p derivation. Examples:

DERI/SLOPE 264/...

DERI/SLOPE/...

DENALT, (OAT)
Density Altitude is calculated from Boom System Static Pressure and OAT according to the process described in Section 6.2. The second entry, (OAT), is as described under the TAS derivation. Example:

DERI/DENALT 29.5/...

DERI/DEN/...

MRAZ

Rotor Azimuth is derived from the rotor azimuth item code as described in Section 6.2. This parameter is displayed as a ramp function ranging from 0.0 to 360 degrees. Either the main or tail rotor azimuth will be displayed according to the 'rotor mode' specified at the beginning of the program run (see Section 5.6). Example:

DERI/MRAZ/...

5.3 INPUT SUBSTEP COMMANDS

The sequence of entries required for the Input substep depends on the preceding Specification and Action substep entries. In addition, this sequence can branch, depending upon the Input substep entries themselves. For example, the first Input substep entry after the sequence

DERI/MRPM/...

is the counter for the input data stream, while for the sequence

ANAL/MMAX/

the first Input substep entry is a source of data. This source could be an individual item code, a group of items specified in the Info File, or a scratch file (SCF1 or SCF2). For each of those possibilities, there is a different sequence of subsequent entries.

There are 19 possible Input substep sequences. These sequences will be displayed in tabular form using short descriptors for the meaning of each entry position. The meaning of every entry descriptor and possible inputs for each are listed below. Descriptors are enclosed in parentheses while actual entries appear in upper case and are enclosed in single quotes.

(Angle) - Rotor azimuth in degrees for single azimuth position input. The start point and length of measured data input time histories is sometimes defined as an integer number of complete rotor cycles. If zero cycles are selected, then a single azimuth position defined by (Angle) is

input. The default (Angle) is -.01 degree. If a nonzero number of cycles is selected, then (Angle) has no meaning and should be defaulted.

(Block Name) - Four-character name of a command sequence. Each command sequence block generated under the command sequencing capability is stored with a unique four-character name which may not begin with any character which would imply a numeric entry. Command sequences are referenced by name for creation, modification, execution, or deletion. There is no default for this entry.

(Comment String) - Sequence of up to 55 characters to be used for additional labeling of plots and printouts. The sequence begins with the first nonblank character after the slash ending the specification substep and ends with the last nonblank character before the slash ending the input substep. The sequence may include blanks or commas but not slashes. Blanks or slashes may not isolate a string which forms a numeric error (e.g., 'THE 3RD FLIGHT'). The comment string may not extend from the end of one line to the beginning of the next but may occupy a full input line after the specification substep was entered on the previous line. The comment is initially blank. Once entered, a comment remains unchanged until a new comment step is executed. The Comment can be set to blanks with an Input substep containing only blanks.

(Counter) - Specifies the unique integer assigned to reference the data from some specific period of time when useful data were taken. Allowed counter values are 1 through 32767. There is no initial default Counter, but, once set, the last Counter specified is the default.

(Cycles) - Integer number of rotor cycles of data input. For certain Specification and Action substep combinations, the period of the input data record is defined as an integer number of contiguous rotor cycles. The number of cycles may be any positive integer up to 1000 or zero as indicated under (Angle). In selecting the number of cycles to process, the user must consider the limited amount of processing storage area in the Processing Program. The default number of cycles is one.

(Dbl Row) - Double-row option selection. When there are two available double-row elements for input, the user must specify whether to use both elements by entering 'TOP' or 'BOTTOM.' The meaning of the latter two possible entries must be interpreted with regard to the type of data being input. The obvious connection of the words 'TOP' and 'BOTTOM' is to the upper and lower surfaces of the rotor blade. However, these entries may also apply, for example, to the amplitude and phase parts of stored Harmonic Analysis output. If 'BOTH' is entered, then both double-row elements will be input if available. If only one element is available, then only that element will be input. If 'TOP' or 'BOTTOM' is entered, then the corresponding double-row element must be present for input. 'BOTH' is the default for this entry.

(Duratn) - Length of input record in seconds. For certain Specification and Action substep combinations, the period of the input data is defined as a length of time. This entry may range in value from 0.0 to 1000 seconds. If '0.0' is entered, then a single data point corresponding to (Time) will be input. When entering (Duratn), the user must consider the available data stored on the Master File and the storage limitations of the program. There is no default for this entry.

(Edit Function) - Command sequence process selected. This entry selects a specific command sequencing function after an 'EDIT' Specification substep. There are three possible entries. If 'NEW' is specified, a new command sequence block is originated and all subsequent commands are copied to that block and not executed until a 'NOEDIT' command is entered. 'CHANGE' specifies that a special processing mode will be entered where existing command sequence blocks can be modified. Special instructions for entering commands in this processing mode can be found in Section 5.7.1. 'DELETE' specifies that the listed command sequence block will be removed from storage and the block name and storage are freed for future use. There is no default for this entry.

(Menu Type) - Kind of Menu to be displayed. The user may have several different kinds of Menu Listings displayed for assistance in generating commands. There are four keywords available to invoke these Menu's. 'DATA' requests a listing of counters which have any data present on the Master File partition that is currently accessed by the Processing Program. 'INFO' requests a listing by name of groups present on the Info File along with the keyword pointers and corresponding item codes listed in the initial section of this file. 'EDIT' requests a listing by name of the command sequence blocks currently on the command sequence file. 'SCRATCH' requests a listing of the current contents of each scratch file. The user could also enter a counter to obtain a listing of item codes present on the Master File partition for that counter along with the length of time the data present represents, the offset from the start of data on the original storage medium to the start of data on the Master File, the breakpoint for any digital filtering applied to the data in storing it on the Master File, and the sample rate for the data as stored on the Master File. There is no default for this entry.

(Name) - Info File Group Name. This entry gives the four-character name of the Info File Group that should be used to specify item codes for data retrieval and geometric information and labels for processing and output. There is no default for this entry.

(Source) - Source of data for processing. This entry specifies whether input data will come from the Master File or a scratch file. If data comes from the Master File, the entry can be either an item code or 'GROUP'. The entry 'GROUP' specifies that an Info File group will be used to define the item codes for input. If data are to come from a scratch file, the user should enter 'SCF1' or 'SCF2' as appropriate. There is no default for this entry.

(Time) - Time offset in seconds. This entry can have two slightly different meanings. When the input record period is defined with (Duratn)

then (Time) is the offset from the start of data on the Master File to the start of data in the input record. When the input record is defined with (Cycles), then (Time) is still an offset from the start of data on the Master File. However, in this case the input data record will start at the beginning of the first complete rotor cycle after this offset. The default for this entry is zero seconds.

(1st Dim) - Definition for the domain of the first dimension variable. When data are extracted from a scratch file, the user has an option. Each data stream to be retrieved may have all available data points extracted or only a single data point may be retrieved from each stream. The keyword 'ALL' specifies that all data points will be retrieved. A numeric entry specifies a single data point. The meaning of the number is dependent upon the subsequent entry (1st Var) for a variable definition. The purpose of this option is to allow the user to select a particular azimuth value, time, frequency, or harmonic in retrieving data from a scratch file. The default for this entry is 'ALL'.

(2nd Dim) - Selection for the domain of the second dimension variable. When data are extracted from a scratch file or from the Master File using an Info File group, the user has an option regarding the second dimension which is chord position in the OLS blade example. Data streams corresponding to all second independent variable positions available are retrieved when the user specifies 'ALL'. Alternatively, the user may specify an element number corresponding to a particular second dimension position for which data streams are available. Element numbers correspond to row sequence positions stored on the specified group of the Info file or on the scratch file. Allowed element numbers are 1 through 64. 'ALL' is the default for this entry.

(3rd Dim) - Selection for the domain of the third dimension which can be radial position for the OLS blade example. This entry option is the same as the second dimension option above except

that an element should correspond to a stored column element number. 'ALL' is the default for this entry.

(1st Var) - Variable definition for a numeric (1st Dim) entry. The meaning of a numeric (1st Dim) entry is established by (1st Var). Data may be stored on the scratch file as a function of time in seconds, frequency in Hz, or Harmonic Number. If the data are stored as a function of Harmonic Number, then 'HARM' or 'IMPL' may be entered to indicate that (1st Dim) specifies a Harmonic Number. If the data are stored as a function of frequency, then 'FREQ' or 'IMPL' may be entered to indicate that (1st Dim) specifies a frequency. When the data are stored as a function of time, then more options are available to the user. 'TIME' or 'IMPL' may be entered to indicate that (1st Dim) specifies a time measured from the beginning of data stored on the Master File partition from which the data were retrieved. Alternatively, 'MRAZ', 'TAS' or 'MRPM' may be entered to indicate that (1st Dim) is an azimuth position, true airspeed, or rotor rpm, respectively. 'IMPL' is the default for this entry.

Table 5 lists all the possible Input substep entry sequences along with all the corresponding Specification and Action substep combinations. The Specification and Action substep combinations for which the Input substep entries are alike are grouped together on this table with each group under the general heading 'From:'. All of the possible Input substep entry sequences for this grouping are listed under the subsequent heading 'Possible Sequences'.

The entries in each of these sequences in Table 5 proceed from left to right in the table. Individual entries are separated by commas. When one sequence branches into two or more sequences, then the descriptor for the entry which determines which branch to take is underlined. Underneath the underlined descriptor, allowed entries for that entry position are listed. These allowed entries are defined in the corresponding descriptor definition as listed above. After a sequence branch, the subsequent entries proceed from left to right at the same level as the entry selection which specified the sequence branch taken.

TABLE 5. INPUT SUBSTEP SEQUENCES

```
From:

ANAL/HARMONIC.../
ANAL/AVERAGE/
ANAL/MMAX/
DERI/CP........./
DERI/BLDIS....../

Possible Sequences:

(Source)

(Item Code),(Counter),(Time)   ,(Cycles) ,(Angle)

 GROUP    , (Name)  ,(Dblrow) ,(3rd Dim),(2nd Dim),(Counter),(Time),(Cycles),(Angle)

SCF1/SCF2  ,(1st Dim)

            (Value)  ,(1st Var),(2nd Dim),(3rd Dim),(Dblrow)

             ALL     ,(2nd Dim),(3rd Dim),(Dblrow)

From:

DISPLAY/
ANAL/FILT..../
ANAL/SPECT.../
ANAL/DAMP..../

Possible Sequences:

(Source)

(Item Code),(Counter),(Time)   ,(Duratn)
```

TABLE 5. (Continued)

```
GROUP      ,(Name)    ,(Dblrow) ,(3rd Dim),(2nd Dim),(Counter),(Time),(Duratn)

SCF1/SCF2  ,(1st Dim)

            (Value)   ,(1st Var),(2nd Dim),(3rd Dim),(Dblrow)

             ALL      ,(2nd Dim),(3rd Dim),(Dblrow)
```

From:

```
DERI/TAS....../
DERI/MRPM/
DERI/MSHP/
DERI/MRAZ/
DERI/MCQ....../
DERI/MCT....../
DERI/DENALT.../
```

Possible Sequences:

(Counter),(Time),(Cycles),(Angle)

From:

```
DERI/MFLO...../
DERI/DFLO...../
```

Possible Sequences:

<u>(Source)</u>

```
 GROUP    ,(Name)    ,(3rd Dim),(2nd Dim),(Counter),(Time),(Cycles),(Angles)
```

TABLE 5. (Continued)

SCF1/SCF2,(1st Dim)

(Value) ,(1st Var),(2nd Dim),(3rd Dim)

ALL ,(2nd Dim),(3rd Dim)

From:

DERI/SLOPE.../
DERI/CN/
DERI/CC/
DERI/C./

Possible Sequences:

(Source)

SCF1/SCF2,(1st Dim)

(Value) ,(1st Var),(3rd Dim)

ALL ,(3rd Dim)

From:

EDIT/

Possible Sequences:

(Edit Function),(Block Name)

TABLE 5. (Concluded)

From:

BUILD/

Possible Sequences:

(Block Name)

From:

EXECUTE/

Possible Sequences:

(Block Name)

From:

MENU/

Possible Sequences:

(Menu Type)

From:

COMMENT/

Possible Sequences:

(Comment String)

For example, suppose a user wishes to derive C_p from absolute pressure data stored on scratch file one for a rotor azimuth position of 90 degrees, for all available chord and radial stations on the top surface of the blade. Assume that the Specification and Action substeps are already entered:

DERI/CP 264 CALC 14.35/

The user can find the appropriate Input substep sequence under the first 'From:' heading in Table 5. The first Input substep entry is indicated by the descriptor (Source) which is underlined to show a branch in the allowable sequences. The possible entries are listed below the underlined descriptor. (Item Code) is an entry descriptor defined in the (Source) explanation. 'GROUP,' 'SCF1,' and 'SCF2' are actual possible entries also defined in the (Source) explanation. Notice that 'SCF1' and 'SCF2' are included on the same line separated by a slash. This slash merely shows that either 'SCF1' or 'SCF2' may be entered, not 'SCF1/SCF2'. Since the input data is stored on scratch file one, the user should enter 'SCF1' for the first entry.

Proceeding directly to the right from the selected entry, the next descriptor, (1st Dim), is underlined to show a second branch in the entry sequence. Underneath this descriptor, the descriptor (Value) and the possible entry 'ALL' are listed. These possible entries are defined under the (1st Dim) explanation. In particular, the user wishes to select the single instant of 90 degrees of azimuth so a (Value) of 90 is entered. Proceeding directly to the right of the (Value) descriptor, the descriptor for the next entry is (1st Var) which should define the meaning of the number 90. The possible entries are not listed under (1st Var) since no branch occurs at this point in the sequence. From the explanation for (1st Var), the proper entry is 'MRAZ' to indicate that 90 is rotor azimuth in degrees.

Again proceeding to the right, the final three entries involve no branches. From the explanation for the descriptors (2nd Dim), (3rd Dim), and (Dblrow), the appropriate entries are 'ALL,' 'ALL,' and 'TOP.' Thus the command developed through the Input substep would be:

DERI/CP 264 CALC 14.35/

SCF1 90 MRAZ ALL ALL TOP/...

with the Disposition substep still to be entered.

5.4 DISPOSITION SUBSTEP COMMANDS

The available sequences of Disposition substep entries are the same for all possible combinations of Specification, Action, and Input substeps, provided that a Disposition substep is required. The user must assume the responsibility to assure that the selected Disposition can be performed given the specified Specification, Action, and Input. When an impossible Disposition is entered, the command step will be accepted for execution and then an error message generated in the processing part of the program. For example, if the user's Specification, Action, and Input substeps result in a process which creates a single time history output and the Disposition calls for a CONTOUR plot, the command step will be accepted but an error message will be generated in processing and no output will appear.

There are ten possible Disposition substep sequences. These sequences will be displayed in the same tabular format as the input substep sequences (see Table 5).

The meaning for every Disposition substep entry descriptor and possible inputs for each are given below:

(Cross) - Tektronix cross-hair cursor activation. This entry determines whether the Tektronix cross-hair cursor will be activated after an X-Y plot is completed to evaluate points on the plot in user coordinates. 'CLOSE' indicates the cursor should not be activated. 'CURSOR' indicates the cursor should be activated. A 'CURSOR' request is ignored in the batch operating mode. 'CLOSE' is the default for this entry.

(Decades) - Decades in log scale. (Decades) is an integer which specifies the number of decades which will be depicted for the logarithmic scale in question. One to six decades are allowed in the X (independent variable) or Y (dependent variable) direction. The default number of decades is three in either direction.

(Format) - Overall format for the output. This entry specifies the type of output. The allowed formats are:

PLOT - X-Y plot with a single curve

MPLOT	-	X-Y plot with one or more curves
APLOT	-	Add a curve to an existing multiple curve X-Y plot
PRINT	-	Print output
CONTOUR	-	Contour plot of a function of two independent variables
SURFACE	-	Surface plot of a function of two independent variables
KEEP	-	Store the process results on the designated scratch file while destroying any data already present on the file.
ADD	-	Store the process results on the designated scratch file in addition to data already present on the file.

There is no default for this entry.

(Indep) - Independent scale variable for X-Y plot. The independent scale variable for an X-Y plot will be completely determined by the input and process performed in generating the output function. Sometimes, however, the computer may hold sufficient information to display different scale variables for a given dimension and the user may choose between these variables. For example, the user must frequently choose between time or azimuth as the first dimension scale variable.

When a single curve is to be plotted, the data must be a function of either the first, second, or third dimension and the independent scale variable specified must be associated with this dimension. When more than one curve is to be plotted in a single command step, the data is a function of two of the three dimensions and the independent variable specified must be associated with the numerically lower dimension present. For example, if the data is a function of radius and chord, then chord must be

the independent scale variable while each curve on the multiple curve X-Y plot represents a different radius position. If the data is a function of the first dimension, then this dimension must be time, harmonic number, or frequency. For frequency, 'FREQ' or 'IMPL' may be entered to indicate a frequency scale. For harmonic number, 'HARM' or 'IMPL' may be entered to indicate a harmonic number scale. For time, 'TIME' or 'IMPL' may be entered for a time scale or 'MRAZ', 'TAS', or 'MRPM' may be entered to specify rotor azimuth, true airspeed, or rpm scales, respectively. If the second dimension is the independent variable dimension, then 'ROW' or 'IMPL' may be entered to indicate a chord position scale. If the 3rd dimension is the independent variable dimension, then this dimension may represent geometric position (radius) or multiple counters. When this dimension represents position or when the dimension is multiple counters with a user-entered number for each counter to be used as a scale, 'COLUMN' or 'IMPL' may be entered. When this dimension represents multiple counters, the user also has the option to enter 'MRAZ', 'TAS', or 'MRPM' to select scales of rotor azimuth, true airspeed, or rotor RPM, respectively. 'IMPL' is always the default for this entry.

(Obs Pos X) - Observer position in the X direction. For SURFACE format plots using rectangular coordinate systems, this entry specifies the observation point for the plot in the first independent variable or X scale. For surface format plots using cylindrical coordinate systems, the X axis is interpreted as the zero-degree direction on the Z (dependent variable) = 0.0 plane. The units of this entry are widths of the displayed function in the X direction. For rectangular coordinate system plots, the X or Y scale is adjusted so that the apparent width of the function in the X direction is approximately the same as the width in the Y direction. The allowed values for this entry are from 1000 to -1000. The default value is 10 for both cylindrical and rectangular coordinate systems.

(Obs Pos Y) - Observer position in the Y direction. This entry is the same as (Obs Pos X), except that position in the second independent variable or Y direction is given. For cylindrical systems, this direction corresponds to 90 degrees. Allowed values are the same as for (Obs Pos X). The default value is 10 for rectangular systems and .3 for cylindrical systems.

(Obs Pos Z) - Observer position in the Z direction. This entry is the same as (Obs Pos X), except that it indicates position in the dependent variable or Z direction. Allowed values are the same as for (Obs Pos X). The default value is 10 for both rectangular and cylindrical coordinate systems.

For observer position specifications in SURFACE plots, the user must assure that the observer position is not between two parts of the surface.

(Scratch) - Scratch file for data storage. This entry specifies the scratch file which will hold the process output as specified by a 'KEEP' or 'ADD' entry. The allowed file names are 'SCF1' or 'SCF2'. There is no default for this entry.

(System) - Coordinate system for three-dimensional plot. This entry specifies the coordinate system to be used in generating a SURFACE or CONTOUR plot. The allowed entries are 'CYLINDRICAL' and 'RECTANGULAR.' The default for this entry is 'CYLI'.

(User Var) - User supplied value to position columns stored on a scratch file. This entry provides the user the capability to generate a scale in the column position direction. For example, the Master File partition might contain several counters representing different rates of descent. The user could process data from each counter and store the results each time using the 'ADD' instruction. Along with each 'ADD', the user could supply the rate of descent for the entry (User Var). These rates of descent

AD-A065 129 BELL HELICOPTER TEXTRON FORT WORTH TEX F/G 1/3
OPERATIONAL LOADS SURVEY - DATA MANAGEMENT SYSTEM. VOLUME I. US--ETC(U)
JAN 79 R B PHILBRICK, A L EUBANKS DAAJ02-77-C-0053
UNCLASSIFIED BHT-299-099-871 USARTL-TR-78-52A NL

2 OF 2
AD
A065129

END
DATE
FILMED
4 -79
DDC

could then be used to position the corresponding column element outputs on a rate of descent scale. Allowed inputs for this entry are 'NONE' or numbers ranging from -10000000 to +10000000. The default for this entry is 'NONE'.

(X-Bottom) - Minimum X scale value. This entry specifies the minimum independent variable or X scale value to depict for a single or multiple curve X-Y plot. The entry 'AUTO' specifies automatic scaling to select this value. A numeric input specifies a user-selected value. If this value is specified in conjunction with a user-specified X-interval, then (X-Bottom) must be an integer multiple of (X-Interval). Allowed input values for (X-Bottom) are 'AUTO' or numeric values between -10000000 and +10000000. The default for this entry is 'AUTO'.

(X-Interval) - X scale interval. This entry specifies the interval between annotated positions on the independent variable axis in independent variable units. The entry 'AUTO' specifies automatic scaling for this value. A numeric input specifies a user-selected value. This number should be a power of ten times one of the numbers: one, two, four, or five. Allowed input values for (X-Interval) are 'AUTO' or numeric values between 0.0 and 10,000. The default for this entry is 'AUTO'.

(X-Var) - X Direction variable. For surface or contour plots, this entry specifies the variable for the first independent axis. This axis could either be 2nd dimension position (which is chord for the OLS blade example), or a time-related variable. The allowed display axis variables are dependent upon the input and process steps. If the data to be plotted is a function of the 2nd and 3rd dimensions, then (X-Var) must be the 2nd dimension which can be called 'ROW' or 'IMPL'. If the data to be plotted is a function of the 2nd or 3rd dimensions and the 1st dimension, then (X-Var) must be time related. If the 1st dimension variable is frequency, then the allowed entries for (X-Var) are 'FREQ' or 'IMPL', which both specify

frequency. If the time related variable is a harmonic number, then the allowed entries are 'HARM' or 'IMPL', which both specify harmonic number. If the time related variable is time, then allowed entries are 'TIME', 'IMPL', 'MRAZ', 'TAS', and 'MRPM'. 'TIME' and 'IMPL' both specify time. 'MRAZ', 'TAS', and 'MRPM' specify rotor azimuth, true airspeed, and rotor speed, respectively. 'IMPL' is always an allowed entry and is the default for this entry.

(Y-Bottom) - Minimum Y scale value. This is the corresponding entry for the dependent variable plot axis as (X-Bottom) is for the independent variable plot axis.

(Y-Interval) - Y scale interval. This is the corresponding entry for the dependent variable plot axis as (X-Interval) is for the independent variable plot axis.

(Y-Var) - Y Direction variable. For surface or contour plots, this entry specifies the variable for the second independent axis. This axis could be the 2nd dimension position (chord in the OLS blade example) or the 3rd dimension position, which can be a radial station in the OLS blade example, or a variable which changes with the counter. If the data to be plotted is a function of the 2nd and 1st dimension, then (Y-Var) must be the 2nd dimension variable which can be called 'ROW' or 'IMPL'. If the data to be plotted is a function of the 3rd dimension and the 2nd or 1st dimension, then (Y-Var) must be associated with the 3rd dimension (column position). The 3rd dimension can correspond to geometric positions (radial position) or a variable which changes with the counter. If the 3rd dimension variable is geometric or if the user has entered column position values in writing the column positions separately to a scratch file, then the appropriate entries are 'COLUMN' or 'IMPL', where the scale will be the column positions specified by the Info file or the user. If the 3rd dimension variable has been generated by storing multiple counters on a scratch file to generate a variation in airspeed or rotor

speed, then the appropriate (Y-Var) entries are 'TAS', 'MRPM', or 'MRAZ' for true airspeed, rotor speed, or rotor azimuth scales, respectively. 'IMPL' is the default for this entry. However, if the 3rd dimension is formed from multiple counters and no user-supplied numbers have been provided as column positions, then the entry 'IMPL' will create an error condition in processing.

(Z-Interval) - Interval between contours. This entry allows the user to specify the interval between levels depicted by a contour plot. These levels correspond to dependent variable or Z values held constant for each individual contour drawn. The interval must be some power of ten times one, two, four, or five. The alternative entry is 'AUTO' which selects autoscaling; 'AUTO' is the default for this entry.

(Z-Val) - Specified contour value. This entry allows the user to enter the minimum function value to be used in the generation of a contour plot. This value must be an integer multiple of the Z-interval or 'AUTO' for autoscaling may be specified. If a specified numeric value is not within the range of the function, then the corresponding contour will not appear in the plot and the value will not be labeled in the annotation area. 'AUTO' is the default for this entry.

Table 6 lists all the possible disposition substep entry sequences. The format for this table is identical to the format for Table 5 except that Specification, Action, and Input substep combinations are not listed since the Disposition substep always begins with the same options.

5.5 EXAMPLES OF COMMAND STEPS

The purpose of this section is to give the user some examples of command steps and sequences of command steps to illustrate the structure, syntax, and allowed entries for user input. Explanations for each command step precede the commands. The commands form a sequence which might represent a typical run using the Interactive Graphics mode of the Processing Program. The assumption is that the required data have already been transferred to the Master File and that the user has already been through the Initialization Phase of program execution (see Section 5.6).

TABLE 6. DISPOSITION SUBSTEP SEQUENCES

```
(Format)

PLOT/MPLOT,(Indep)  ,(Cross)  ,(Y-Interval)
                               AUTO/(Value),(Y-Bottom)  ,(X-Interval)
                                                         AUTO/(Value),(X-Bottom)
                                                         LOG         ,(Decades)
                               LOG         ,(Decades)   ,(X-Interval)
                                                         AUTO/(Value),(X-Bottom)
                                                         LOG         ,(Decades)

APLOT      ,(Indep)

PRINT      ,(Indep)

CONTOUR    ,(System) ,(X-Var)  ,(Y-Var)     ,(Z-Interval),(Z-Val)

SURFACE    ,(System) ,(X-Var)  ,(Y-Var)    ,(Obs Pos X) ,(Obs Pos Y),(Obs Pos Z)

KEEP       ,SCF1/SCF2(User Var)

ADD        ,SCF1/SCF2,(User Var)
```

At the beginning of the run, the user checks the data present on the Master File partition with the MENU instruction. First, the counters present on the partition are checked:

```
 NEW STEP
MENU/DATA/
```

The counters 685, 686, 687, 688, 689 and 690 are listed on the screen.

The user is interested in counter 690, so a listing of the items present for that counter is requested.

```
 NEW STEP
MENU/690/
```

Figure 16 shows the output for this step.

Menu listing control (see Paragraph 5.7.4) is not required because fewer than 50 lines are generated in the menu.

The user now obtains a True Airspeed plot for counter 690. Both the static pressure and OAT item codes are present for counter 690, so the corresponding entries are defaulted to CALC. The user requests three rotor cycles of airspeed data beginning with the first complete cycle after the start of data for the counter. An X-Y plot output is requested with time as the independent axis (default = IMPL), no cross-hair cursor activation, and autoscaling for both the 'X' and 'Y' axes.

```
 NEW STEP
DERI/TAS/690 0 3/PLOT/
 EXECUTING
```

Figure 17 shows the True Airspeed plot output generated on the right-hand side of the screen. The Tektronix screen will always be cleared before a plot is generated except when the APLOT command is used.

The user now enters the command sequence BUILD mode so that subsequent steps can be recorded and executed in the batch mode. The name of the command sequence to be generated is 'PR90'.

```
 NEW STEP
BUILD/PR90/
```

NEW STEP STEP CPU 0.23 SEC TOTAL CPU 1.2 SEC
MENU/ 690.000/

ITEM CODE LIST

ITEM	SEC	FILT	OFF	RATE	ITEM	SEC	FILT	OFF	RATE
P002	1.0	50	0.2	256	P030	1.0	50	0.2	256
P157	1.0	100	0.2	512	P158	1.0	100	0.2	512
P159	1.0	100	0.2	512	P160	1.0	100	0.2	512
P161	1.0	100	0.2	512	P162	1.0	100	0.2	512
P163	1.0	100	0.2	512	P164	1.0	100	0.2	512
P165	1.0	100	0.2	512	P166	1.0	100	0.2	512
P173	1.0	100	0.2	512	P174	1.0	100	0.2	512
P175	1.0	100	0.2	512	P177	1.0	100	0.2	512
P178	1.0	100	0.2	512	P179	1.0	100	0.2	512
P181	1.0	100	0.2	512	P182	1.0	100	0.2	512
P194	1.0	100	0.2	512	P195	1.0	100	0.2	512
P738	1.0	100	0.2	512	P740	1.0	100	0.2	512
P757	1.0	100	0.2	512	P813	1.0	100	0.2	512
P814	1.0	100	0.2	512	P815	1.0	100	0.2	512
P828	1.0	100	0.2	512	P829	1.0	100	0.2	512
P831	1.0	100	0.2	512	P836	1.0	100	0.2	512
P837	1.0	100	0.2	512	P838	1.0	100	0.2	512
P839	1.0	100	0.2	512	P840	1.0	100	0.2	512
P841	1.0	100	0.2	512	P843	1.0	100	0.2	512
P844	1.0	100	0.2	512	P845	1.0	100	0.2	512
P852	1.0	100	0.2	512	P853	1.0	100	0.2	512
P854	1.0	100	0.2	512	P855	1.0	100	0.2	512
P856	1.0	100	0.2	512	P857	1.0	100	0.2	512
P858	1.0	100	0.2	512	P859	1.0	100	0.2	512
P860	1.0	100	0.2	512	P861	1.0	100	0.2	512
P868	1.0	100	0.2	512	P869	1.0	100	0.2	512
P870	1.0	100	0.2	512	P871	1.0	100	0.2	512
P873	1.0	100	0.2	512	P874	1.0	100	0.2	512
P875	1.0	100	0.2	512	P876	1.0	100	0.2	512
P877	1.0	100	0.2	512	P884	1.0	100	0.2	512
P908	1.0	100	0.2	512	P909	1.0	100	0.2	512
P919	1.0	100	0.2	512	P920	1.0	100	0.2	512
P921	1.0	100	0.2	512	P926	1.0	100	0.2	512
P927	1.0	100	0.2	512	P928	1.0	100	0.2	512
P941	1.0	100	0.2	512	P942	1.0	100	0.2	512
P943	1.0	100	0.2	512	P957	1.0	100	0.2	512
P958	1.0	100	0.2	512	P959	1.0	100	0.2	512
P973	1.0	100	0.2	512	P974	1.0	100	0.2	512
P975	1.0	100	0.2	512	P989	1.0	100	0.2	512
P990	1.0	100	0.2	512	P991	1.0	100	0.2	512
R992	1.0	-1	0.2	512	T004	1.0	50	0.2	256

Figure 16. Menu of item codes for counter 690.

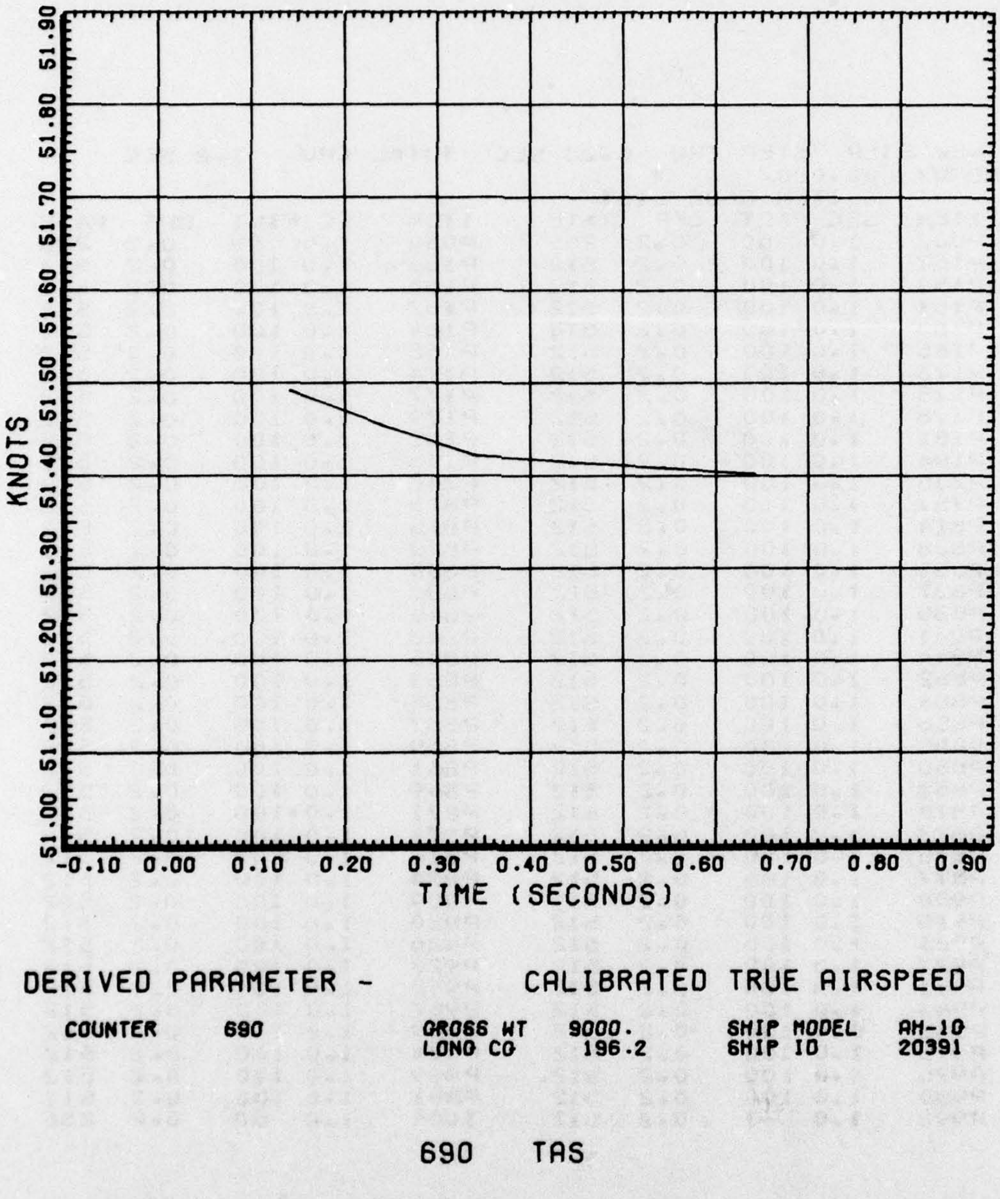


Figure 17. True airspeed for counter 690.

The user now enters a comment to be used as additional labeling for the plots which will be generated in this session.

```
 NEW STEP
COMM/LEVEL FLIGHT AT 50 KNOTS/
```

The user saves the comment step on the command sequence block.

```
 NEW STEP
SAVE/
```

The user now begins the analysis procedures required to produce C_n data from Blade Absolute Pressure measurements. First, the measurements from all the absolute pressure sensors on the OLS blade (white blade) are cycle averaged over three cycles. The Info file is used to supply the proper item codes. Geometric position information and labels are also extracted from the Info file. The results of the process are written to scratch file one (SCF1).

```
 NEW STEP
ANAL/AVERAG/GROUP S2PP BOTH ALL ALL 690 0 3/KEEP SCF1/
 EXECUTING
```

The EXECUTING message informs the user that processing has begun.

The 'NEW STEP' message (below) indicates the step has worked without any error being detected. The user records this step on the command sequence block.

```
 NEW STEP
SAVE/
```

The user now requests a C_p derivation from the cycle-averaged absolute pressure measurements stored on SCF1. A blade radius of 264 inches is specified, and the entries for static pressure and OAT are allowed to default to CALC. The input for this derivation is the data stored on SCF1. The 1st dimension (time) span, the row and column (chord and radius) element, and the double-row element (upper and lower surface) selections are allowed to default to ALL, ALL, and BOTH so that all the data stored on the scratch file is used. The output of the derivation is stored on SCF2.

```
 NEW STEP
DERI/CP 264/SCF1/KEEP SCF2/
 EXECUTING
```

This step appears to work properly, so the user records the step on the command sequence block.

```
 NEW STEP
SAVE/
```

Now the user requests a C_n derivation from the C_p values stored on SCF2. As with the C_p derivation, defaults are used to specify that all of the data on the scratch file are used for input. The output of the derivation is stored on SCF1.

```
 NEW STEP
DERI/CN/SCF2/KEEP SCF1/
 EXECUTING
```

The step appears to work properly, so the user records the step on the command sequence block.

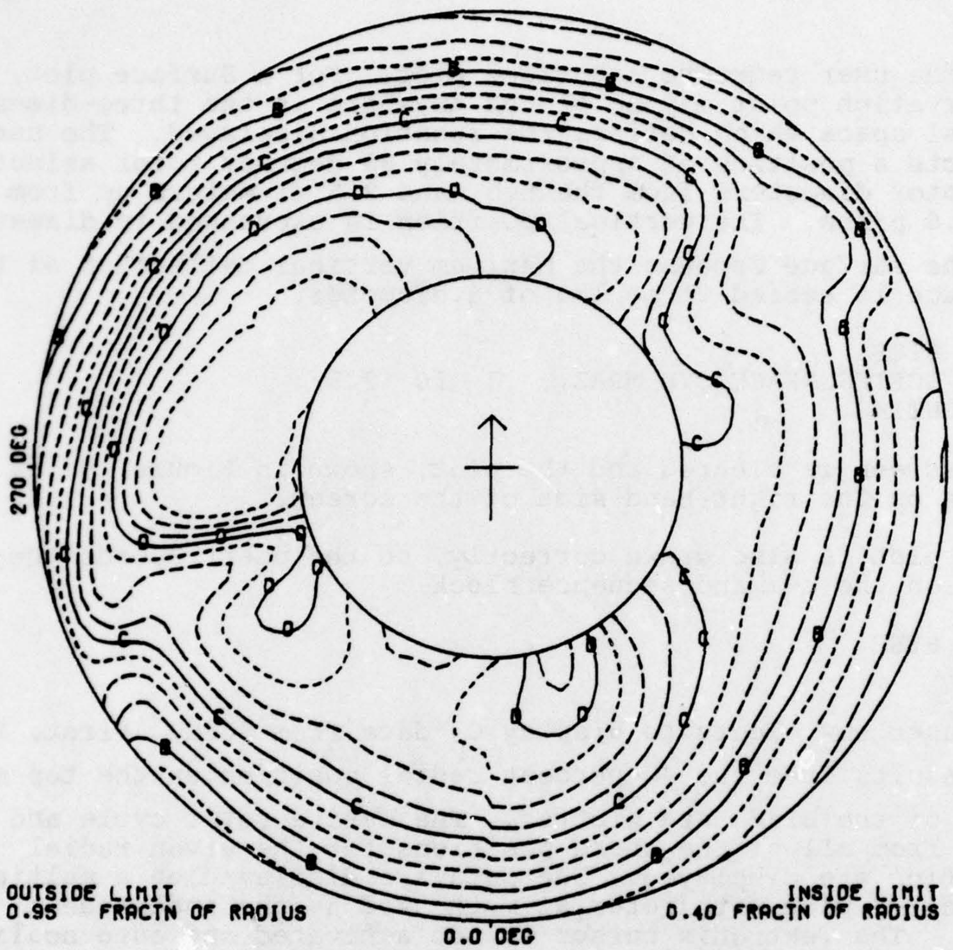
```
 NEW STEP
SAVE/
```

Now the user will display the derived C_n values which have been calculated. The C_n integration has reduced the number of independent variables being carried from three to two: radius and time or rotor azimuth. All of the C_n values stored on SCF1 can be represented in a single three-dimensional plot. First, the user selects a Contour plot of the data. For the input from SCF1, the defaults which specify that all of the data will be used are correct. The Disposition substep specifies a contour plot using a cylindrical coordinate system with the first dimension variable defined as azimuth. For a Surface or Contour plot with a cylindrical format, the first dimension variable must be specified as rotor azimuth. The default entry, 'IMPL', would specify time as the first dimension. The other Disposition substep entries are allowed to default where 'IMPL' specifies column position (radius) as the second independent variable for the plot, and 'AUTO' specifies auto scaling for both of the scaling entries.

```
 NEW STEP
DISP/SCF1/CONTOUR CYL MRAZ/
 EXECUTING
```

The screen is cleared and the plot, shown in Figure 18, is drawn on the right-hand side of the screen.

The plot is drawn correctly, so the user records the step on the command sequence block.



LEVEL FLIGHT AT 50 KNOTS

DERIVED PARAMETER - NORMAL FORCE COEFFICIENT

COUNTER 690 GROSS WT 9000. SHIP MODEL AH-1G
LONG CG 196.2 SHIP ID 20391

A 0.10 CN
B 0.30 CN
C 0.50 CN
D 0.70 CN

BHT.USARTL OLF/DMS (VERS 1.31 - 06/23/78) 06/28/78

Figure 18. C_n contour plot for counter 690.

NEW STEP
SAVE/

Now the user requests a Surface plot. For a Surface plot, the observation point may be placed anywhere in the three-dimensional space which defines the function displayed. The user selects a position of approximately 88 degrees rotor azimuth, 10 rotor diameters from the hub, and 2.5 diameters up from the C_n=0.0 plane. The vertical position is expressed in diameters of the surface because the maximum vertical deflection of the surface is scaled to be 1/4 of a diameter.

```
 NEW STEP
DISP/SCF1/SURFACE CYL MRAZ,, .3  10  2.5/
 EXECUTING
```

The screen is cleared and the plot, shown in Figure 19, is drawn on the right-hand side of the screen.

This plot is also drawn correctly, so the user records the step on the command sequence block.

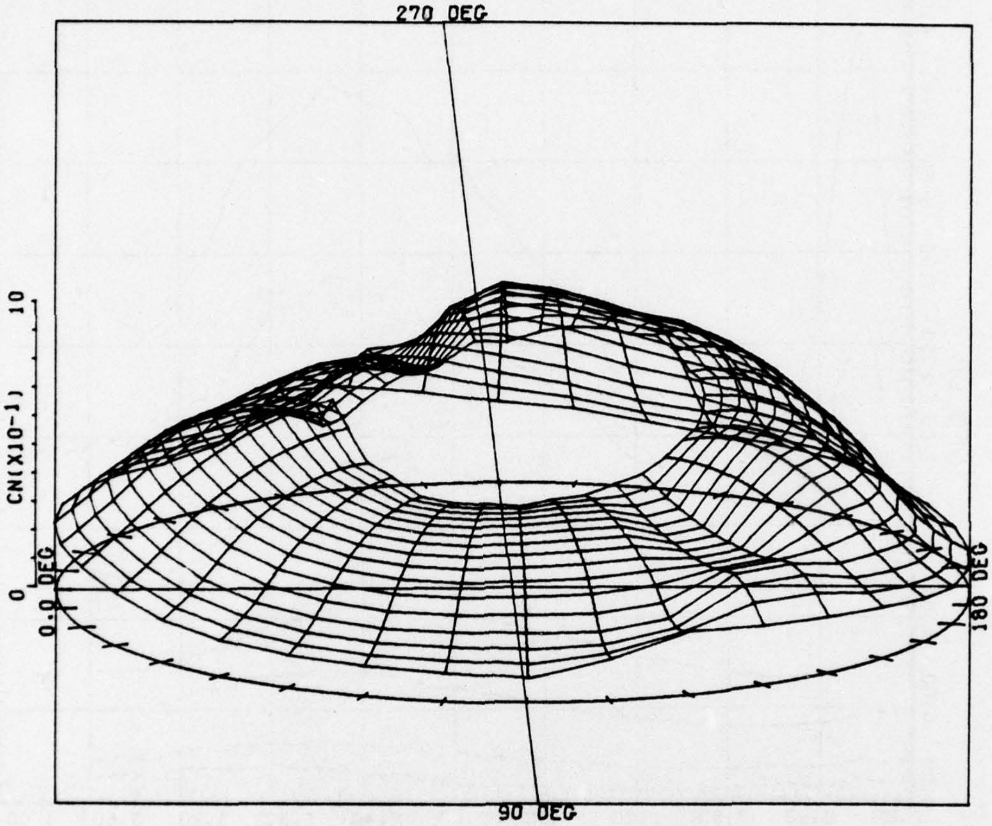
```
 NEW STEP
SAVE/
```

The user now elects to display C_p data from SCF2. First, the C_p results from the 40-percent radial position on the top surface of the blade are plotted. The entire rotor cycle and the data from all of the chord positions for the given radial position are displayed. The data are displayed on a multiple curve X-Y plot with rotor azimuth used as the independent axis. The Tektronix cursor is not activated and auto scaling is used for both axes.

```
 NEW STEP
DISP/SCF2 ALL ALL 1 TOP/MPLOT MRAZ/
 EXECUTING
```

The screen is cleared and the plot, shown in Figure 20, is drawn on the right-hand side of the screen. The user decides that an incremental pen plotter copy of this plot is unnecessary so a SAVE step is not entered.

For a different perspective on the C_p function, the user now plots the upper surface, 40 percent radius data, using chord as the independent variable. The curve is generated for several rotor azimuth positions using the 'APLOT' output option. The first curve, at 10 degrees of azimuth, is drawn using MPLOT with ROW defined as the independent variable and the



LEVEL FLIGHT AT 50 KNOTS

DERIVED PARAMETER - NORMAL FORCE COEFFICIENT

COUNTER	690	GROSS WT	9000.	SHIP MODEL	AH-1G
		LONG CG	196.2	SHIP ID	20391

ANGULAR INCREMENT 10 DEG
RADIAL QUANTITY FRACTN OF RADIUS
MAX RADIUS 0.955
RADIAL INCREMENT 0.0370

BHT.USARTL OLS/OMS (VERS 1.31 - 06/23/78) 06/28/78

Figure 19. C_n surface plot for counter 690.

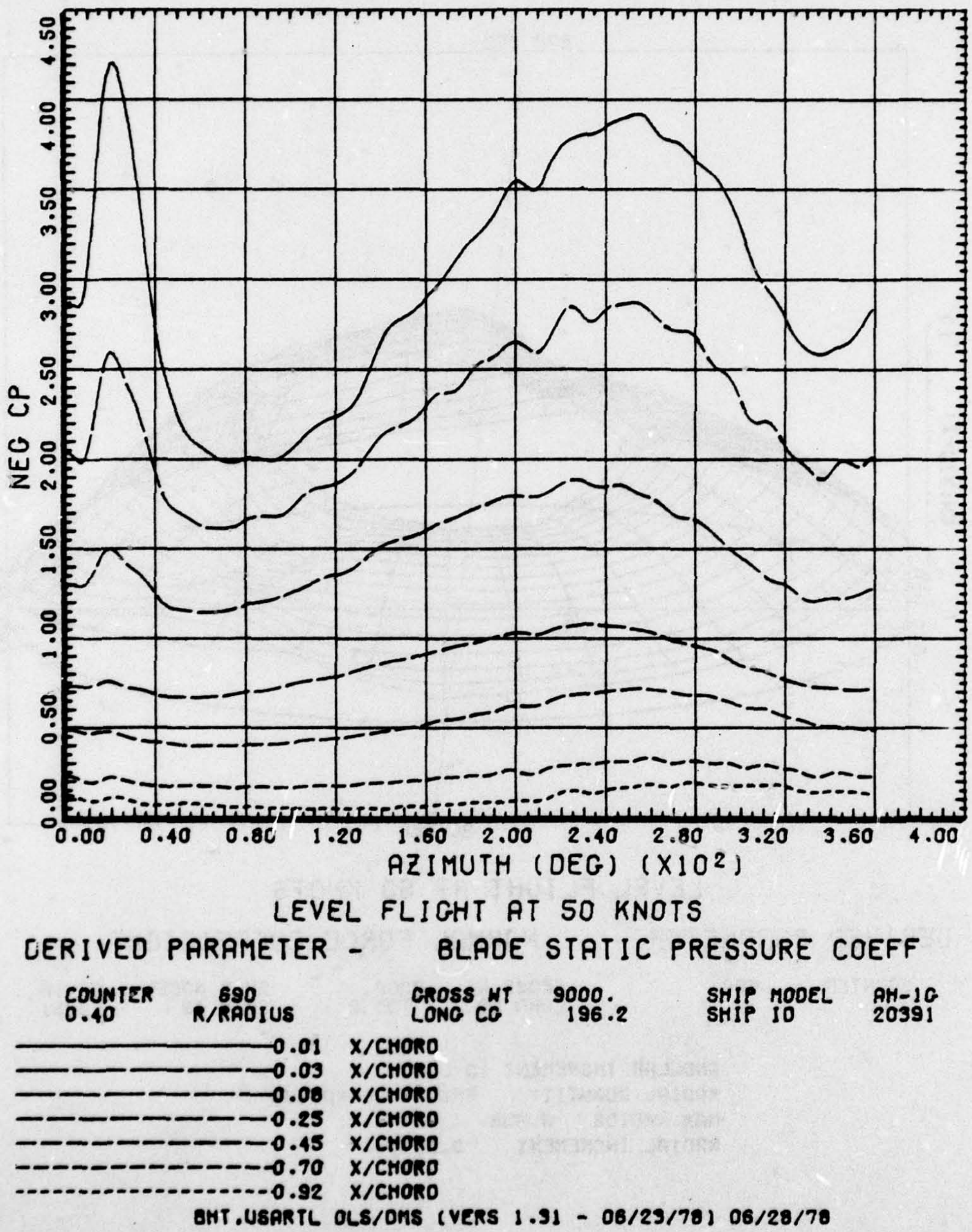


Figure 20. C_p multiple curve plot, counter 690, 40 percent radius, upper surface.

Tektronix cursor is not used. The dependent axis scale must be defined by the user to accommodate higher C_p values that will occur at 20 degrees. The user selects a 'Y' scale interval of .5 and a minimum 'Y' scale value of 0.0. Autoscaling is used for the 'X' axis.

```
 NEW STEP
DISP/SCF2 10 MRAZ ALL 1 TOP/
MPLOT ROW CLOS .5 0/
 EXECUTING
```

The screen is cleared and the plot, shown in Figure 21, is drawn on the right-hand side of the screen. This command step is not saved.

Now the user adds the 20-degree azimuth position for the same radial station. The curve is plotted on the same grid using the same scale. The independent variable is defaulted to 'IMPL' since ROW is the natural scale created.

```
 NEW STEP
DISP/SCF2 20 MRAZ ALL 1 TOP/APLOT/
 EXECUTING
```

A dashed curve is added to the plot on the right-hand side of the screen so that it now appears as shown in Figure 22.

A curve for the 30-degree azimuth position is now added to the plot.

```
 NEW STEP
DISP/SCF2 30 MRAZ ALL 1 TOP/APLOT/
 EXECUTING
```

Another dashed curve is added to the plot on the right-hand side of the screen so that it now appears as shown in Figure 23. None of the commands which create this plot have been saved on the Command Sequence block.

The user now terminates the Command Sequence BUILD mode.

```
 NEW STEP
NOEDIT/
```

Command sequence block, 'PR90', is now available for execution at a later time in the Batch Mode. The user terminates the program session.

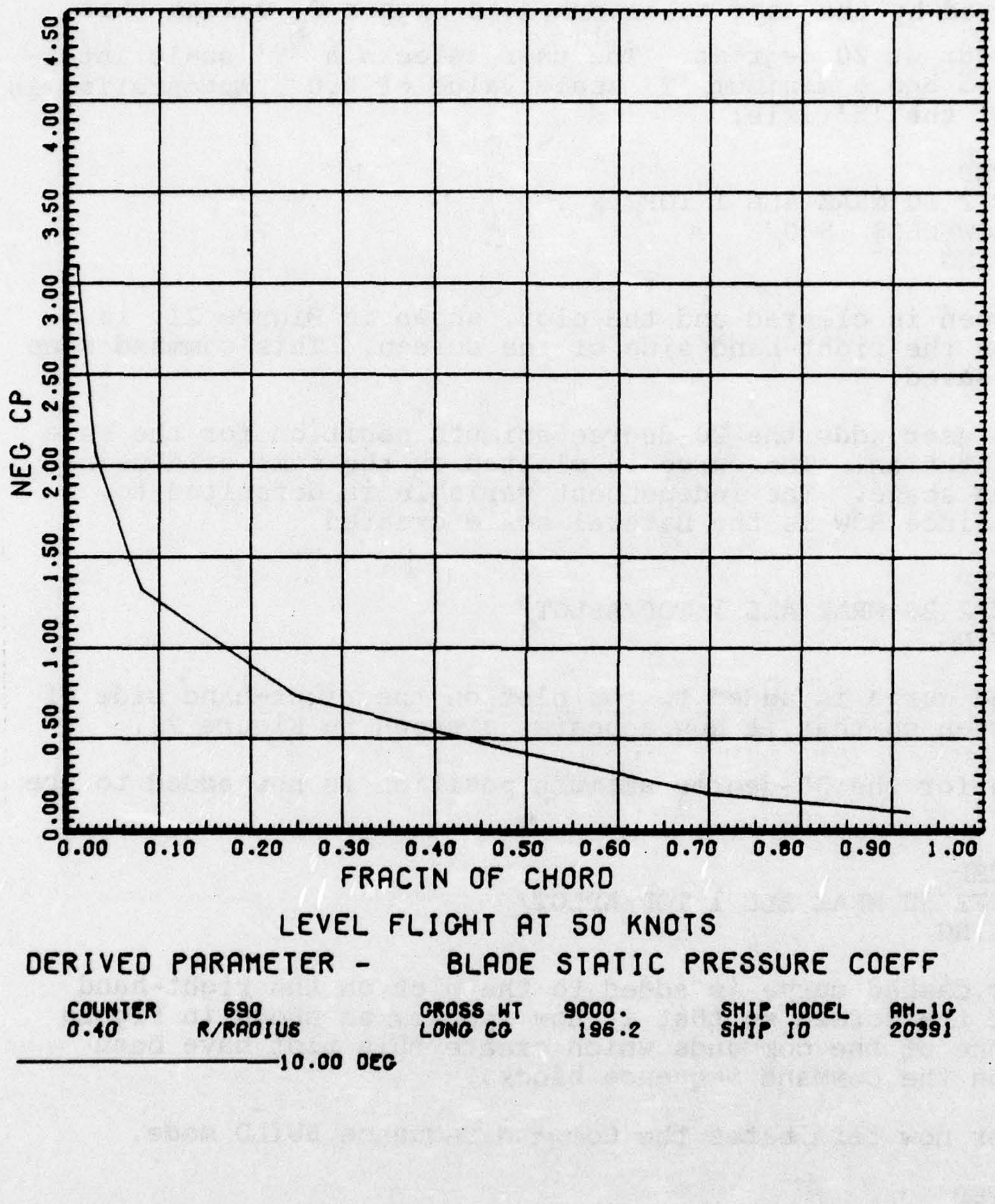


Figure 21. C_p versus chord at 10 degrees azimuth.

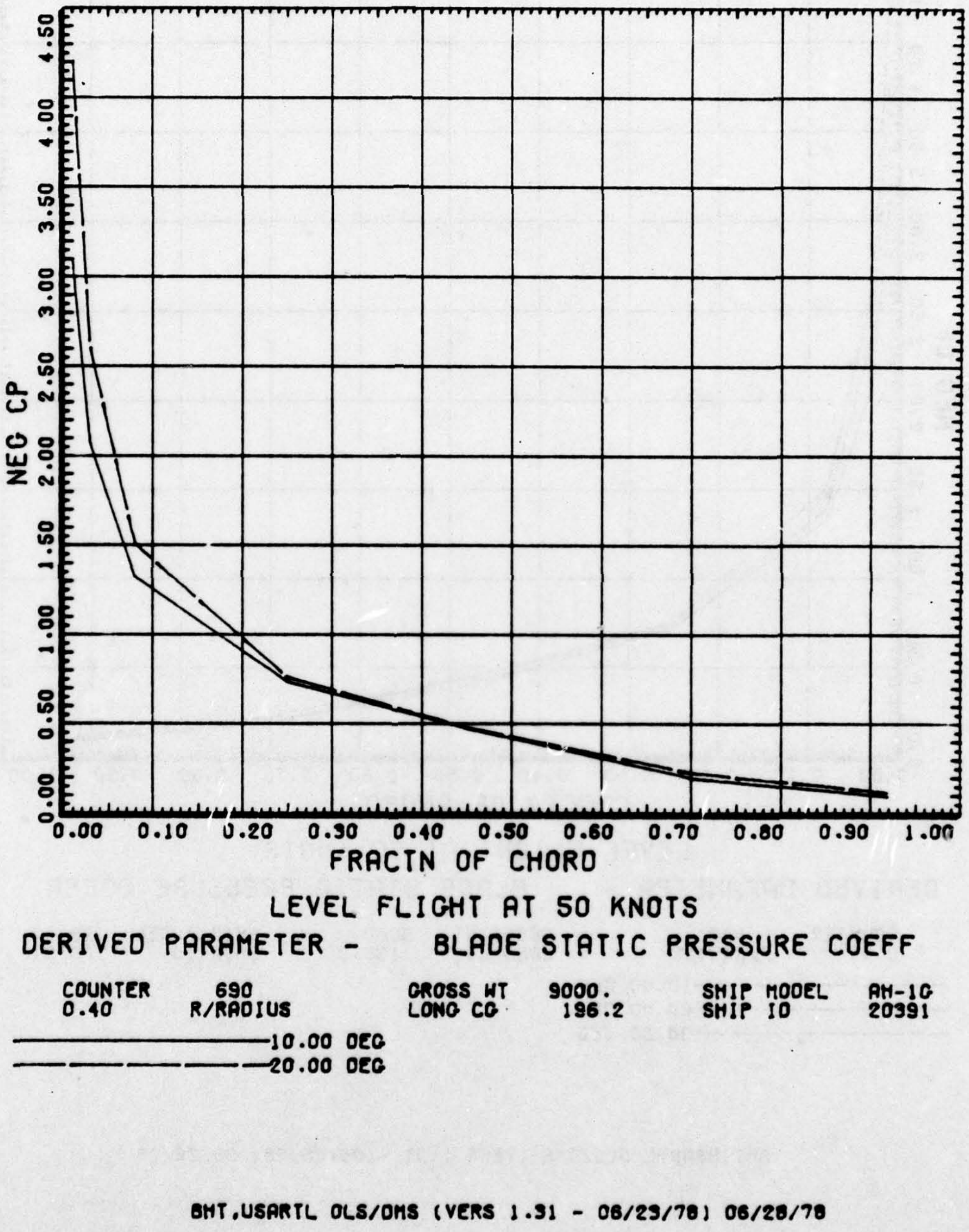


Figure 22. C_p versus chord for two azimuth positions.

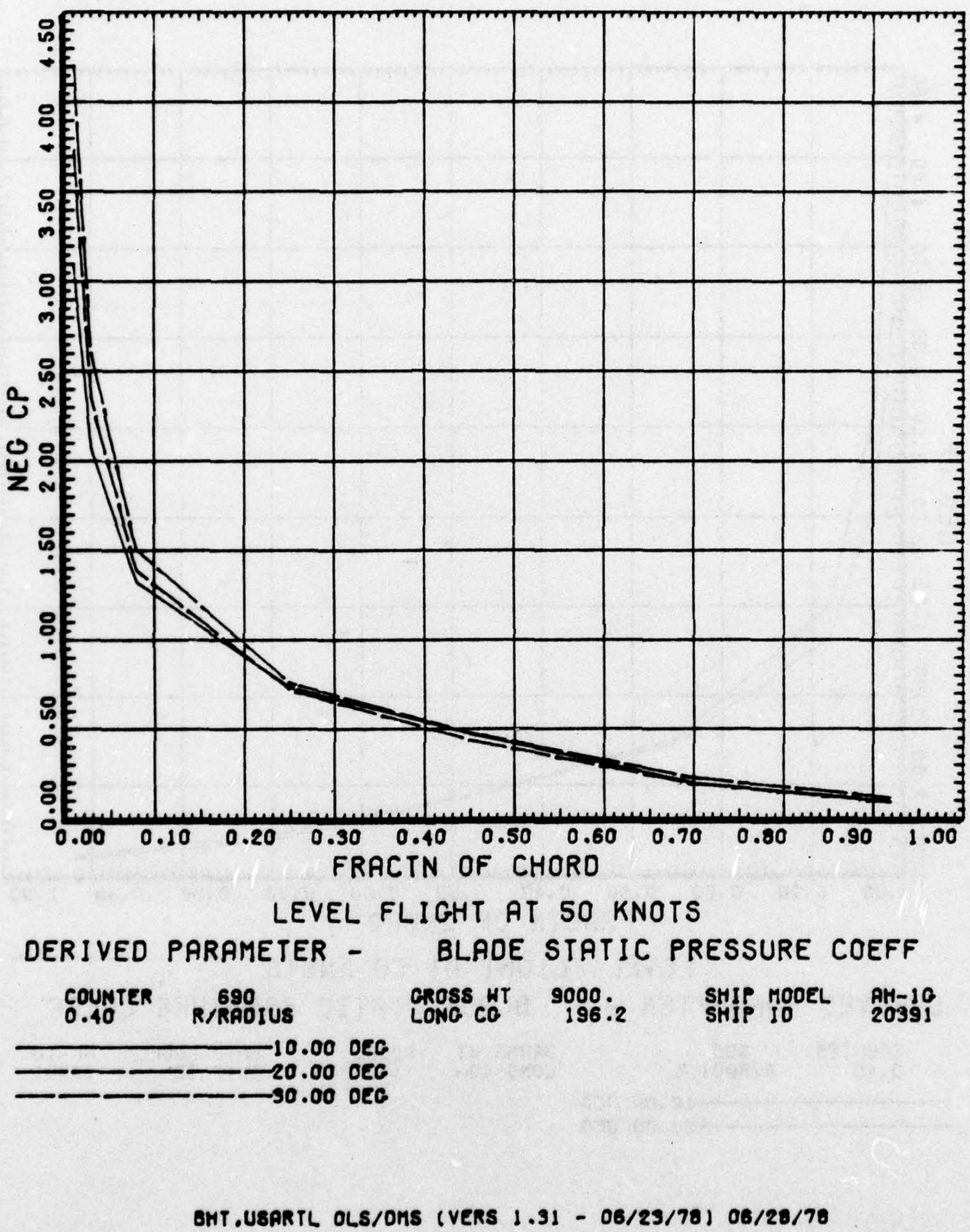


Figure 23. C_p versus chord for three azimuth positions.

NEW STEP
TERM/
RUN TERMINATES

The example sequence is now complete.

5.6 PROGRAM INITIALIZATION PHASE

Immediately upon the beginning of program execution and before the command steps described in Sections 5.1 through 5.5 can be accepted, the computer executes an initialization phase. Two types of activity occur during the initialization phase: (1) the program elicits certain information from the user regarding the run; and (2) the program initializes or verifies certain disc files. This phase occurs in all three operating modes: Batch, Interactive, and Interactive Graphics. In the Batch mode, the user must assure that the proper entries for this phase are present at the beginning of the user input sequence. When the program is running in the Batch mode, an erroneous input for the initialization phase terminates program execution.

The very first information requested by the Processing Program is the mode of operation. The program prints the following message:

BHT/USARTL OLS/DMS (VERS 1.32 06/23/78)

ENTER OPERATING MODE:
1 = BATCH
2 = INTERACTIVE (NO PLOTS)
3 = INTERACTIVE GRAPHICS (TEKTRONIX NEEDED)

The user should enter the number corresponding to the proper operating mode. The program will repeat the selected number back to the user (in the Batch mode, only the echoed number will appear in the printout).

For any of the inputs to the Program Initialization Phase, the user may substitute the alternative keywords 'RESTART' or 'END'. 'RESTART' will cause the program to return to the beginning of the initialization sequence where the message listed above is printed. The user will then have the opportunity to repeat the entire sequence of Initialization Phase entries. 'END' will cause the program to terminate.

Following the entry of a number to specify the operating mode, the program prints a list of operating settings and keywords for modification of these settings. This list is shown at the

top of Figure 24. The user is then prompted to enter 'YES' to accept the existing settings or one of the listed keywords to modify a setting. When there is a selection between two keywords to indicate one setting, the keyword entry alone is sufficient to define the setting and modification. When a single keyword is available to select a setting, then, after this keyword is entered, the computer will prompt the user for a numeric entry. This number may be entered in free field format using the rules for numeric entries presented in Section 4.1. However, the number should not be entered on the same line as the keyword.

Figure 24 shows two examples of modification of these settings. First, the user enters 'STEP' to specify that the CPU time required to execute each step will be listed at the beginning of the following step. No numerical input is required to modify this setting. The computer immediately lists the modified setting and then prompts the user to enter 'YES' to accept the new settings or a new keyword to further modify the run settings.

Continuing with Figure 24, the user enters 'BLOCKS' to specify a change in page size for printed output. The computer prompts the user for the number of five line blocks to list on a printed page. The user responds with '8' for eight blocks. The computer now prints a line giving the modified setting and prompts the user to enter 'YES' or a new keyword. The user enters 'YES' to terminate action on the run settings and the computer prints the message: 'INITIALIZING SCRATCH'. This message is informative and no input from the user is required at this time.

The following list gives the meaning for each of the user selectable run settings listed in Figure 24.

<u>TERMINAL DATA RATE</u> - This entry is significant only when the Interactive Graphics mode is used with a Tektronix terminal. The number entered should give the data communication rate between the terminal and the computer interface in characters per second. For example, if the data communications rate were 2400 baud, then the proper number for this entry is 240 characters per second, which is the default value listed in Figure 24. Normally this default is set for the proper rate at a given installation. The keyword to modify this setting is 'LINE'.

```
BHT,USARTL OLS/DMS (VERS 1.31 - 06/23/78)

ENTER OPERATING MODE:
  1 = BATCH
  2 = INTERACTIVE (NO PLOTS)
  3 = INTERACTIVE GRAPHICS (TEKTRONIX NEEDED)
3
3

     RUN SETTINGS:                                        KEYWORD
  TERMINAL DATA RATE 240 CHARACTERS/SECOND                 'LINE'
  ROTOR MODE 'MAIN'                                 'MAIN','TAIL'
  PLOT GRID MODE  'GRID'                          'GRID','NOGRID'
  PLOT TICS MODE  'TICS'                          'TICS','NOTICS'
  PLOT FRAME WIDTH  8.50 INCHES                            'PWID'
  OPERATOR PEN PLACEMENT IN 'X' -1.50 INCHES               'PENX'
  OPERATOR PEN PLACEMENT IN 'Y'  0.50 INCHES               'PENY'
  PRINT BLOCKS OF 5 LINES/BLOCK   6/PAGE                 'BLOCKS'
  FILES SCF1 AND SCF2 SIZE  300 RECORDS                'FILESIZE'
  CPU SECONDS TO TRIGGER WARNING 30.00                     'WARN'
  STEP EXECUTION TIMES WILL NOT BE PRINTED        'STEP','NOSTEP'

  ENTER 'YES' TO ACCEPT THESE VALUES OR
  A KEYWORD TO MODIFY A SETTING.

STEP
  STEP EXECUTION TIMES   WILL   BE PRINTED        'STEP','NOSTEP'

  ENTER 'YES' TO ACCEPT THESE VALUES OR
  A KEYWORD TO MODIFY A SETTING.
BLOCKS
 ENTER NUMBER OF 5 LINE BLOCKS PER PRINTOUT PAGE
8
  PRINT BLOCKS OF 5 LINES/BLOCK   8/PAGE                 'BLOCKS'

  ENTER 'YES' TO ACCEPT THESE VALUES OR
  A KEYWORD TO MODIFY A SETTING.
YES
 INITIALIZING SCRATCH
```

Figure 24. Initialization Phase run setting modification sequence (example).

<u>ROTOR MODE</u> - This entry allows the user to specify which rotor will be used to specify rotor azimuth and rotor cycles for input specification, processing, and output scales. 'MAIN' specifies the main rotor, and 'TAIL' the tail rotor. 'MAIN' is the default rotor selection.

<u>PLOT GRID MODE</u> - This entry instructs the computer whether to draw a grid on X-Y plots with linear scales. The keyword 'GRID' specifies that a grid line will be drawn for each annotated position on a linear scale. 'NOGRID' specifies that no grid lines will be drawn for linear scales. Grid lines will always be drawn for log scales regardless of 'GRID' or 'NOGRID' selection. 'GRID' is the default entry for this selection.

<u>PLOT TIC MODE</u> - This entry instructs the computer whether to draw tic marks on linear scales of X-Y plots. The keyword 'TICS' specifies that tics will be drawn while 'NOTICS' specifies that no tic marks will be drawn. This entry does not affect tic marks drawn for log scales. 'TICS' is the default entry for this selection.

<u>PLOT FRAME WIDTH</u> - This entry controls the spacing between plot frames when an incremental plotter is used. The keyword 'PWID' specifies that the user will enter the horizontal width allowed for each frame of an incremental plot including the frame itself and the spacing between frames. This entry has no effect on the size of an actual plot frame. The value entered for the frame width must be between 7.5 and 20.0 inches. The default for this entry will vary from installation to installation.

<u>OPERATOR PEN PLACEMENT IN 'X'</u> - This entry informs the computer of the initial incremental plotter pen placement in the 'X' direction relative to the <u>standard pen starting position</u>. The <u>standard pen starting position</u> is one-half inch to the right and 3.5 inches down from the lower left-hand corner of the first plot, assuming that the first plot is an X-Y plot. For the Houston Instruments DP-1, using preperforated paper, this <u>standard pen starting position</u> is one-half inch up from the lower perforation and one-half

inch to the right of a perforation separating two sheets of paper. The operator pen placement specifications allow the user to adjust the plots for differences in initial placement of the pen. For the numbers used in Figure 24, the operator pen placement should be one inch above the lower perforations and one inch to the left of the perforations separating two pages. Allowed 'X' pen placement specifications are -7.5 to 7.5. The keyword for pen placement in the 'X' direction is 'PENX'. The default for this value will change from installation to installation.

<u>OPERATOR PEN PLACEMENT IN 'Y'</u> - This entry informs the computer of the initial incremental plotter pen placement in the 'Y' (vertical) direction relative to the <u>standard pen starting position</u>. See the previous entry description for a definition of this position. The allowed 'Y' pen placement specifications are -7.5 to 7.5. The keyword for pen placement in the 'Y' direction is 'PENY'. The default for this value will change from installation to installation.

<u>PRINT BLOCKS OF 5 LINES/BLOCK</u> - This entry tells the computer how many blocks to print on a page of output. A block consists of five lines of printed data and one blank line. Thus the total number of lines on a page is this entry times six plus a few header lines. The keyword for this entry is 'BLOCKS'. Allowed values are 1 to 1000. Defaults for this entry change depending upon the mode of operation.

<u>FILES SCF1 AND SCF2 SIZE</u> - This entry specifies the size, in records, of each of the scratch files, SCF1 and SCF2. The specified size also applies to the temporary scratch file which is otherwise invisible to the user. The keyword for this entry is 'FILESIZE'. Generally the default value for this specification will correspond to the size of the scratch area provided in the JCL so that the JCL must be modified to allow this number to be increased. If a number of records which exceeds the available disc space is specified, an abnormal termination of the program run will occur. Allowed values

range from 0 to 9999. The default for this entry will change from installation to installation.

CPU SECONDS TO TRIGGER WARNING - This entry specifies the number of Central Processor Unit (CPU) seconds which the program will use before generating time warning messages. When the specified time is exceeded, the warning messages will appear before each 'NEW STEP' message and list the CPU seconds used by the program up to that time. The keyword for this entry is 'WARN'. Allowed values for this entry are 0.001 through 999 seconds. The default for this entry changes from installation to installation.

STEP EXECUTION TIMES PRINTING - This entry instructs the computer whether to print the CPU execution time for each command step during the current run. These times are printed along with the 'NEW STEP' message which prompts the user for the subsequent step input. The keyword 'STEP' specifies that these CPU times will be printed while 'NOSTEP' specifies that the times will not be printed. The default for this entry is 'NOSTEP'.

When the message 'INITIALIZING SCRATCH' appears on the screen, the computer proceeds to initialize the scratch files SCF1, SCF2, and the temporary scratch file. In addition, the Info file and the Command Sequence file are checked for validity. This process may require as much as a few minutes if the computer is extrememly busy or if the scratch files are exceedingly long. When the initialization is complete, the computer prints the following prompting message:

ENTER PARTITION NAME

The user should enter the name of the Master File partition containing the appropriate data for the run. When the corresponding partition is found, the computer will immediately display the prompting message, 'NEW STEP', and the Initialization phase will be complete. If the partition name which has been entered is not found, then the computer will display the message

PARTITION NAME NOT FOUND!!
ENTER PARTITION NAME

and the user will be able to enter a corrected partition name.

5.7 USER INPUT DURING COMMAND STEP EXECUTION

Normally, all user instructions are entered in the form of command steps before the execution of those steps, and no additional instructions are entered until the step execution is complete. Occasionally, however, the user must enter instructions during the execution of a step. These occasions are restricted to the Interactive and Interactive Graphics modes of operations.

5.7.1 The 'Change' Mode of the 'EDIT' Specification

The EDIT/CHANGE command provides the capability to perform the following operations on a recorded sequence, or block, of command steps: list the sequence, change lines in the sequence, insert lines in the sequence, and delete lines from the sequence. Certain other functions are provided in this mode to facilitate use of these capabilities. These functions include renumbering the block of command steps, listing available commands, listing certain lines from the block, and exiting the change mode.

When the Change mode is entered, a completely new structure and syntax for user instructions is introduced. Six keyword commands are recognized: $A, $C, $L, $N, $E and $?. Some of these keywords may be followed by one or two line numbers. Line numbers may be determined from a listing ($L) of the block. Following is a list of keyword commands with '( )' used to signify possible line number(s). The numbers themselves should not be enclosed in parentheses.

$A ( ) - Insert one or more lines. When this command is entered, the user may enter one or more lines under the command line. These lines will be inserted just after the line specified by ( ). The insertion lines end when a command keyword is entered.

$C ( ) ( ) - Change or delete lines. When this command is entered, one or more lines are deleted and zero or more lines may be entered to replace them. If one line number is specified, then that line is deleted. If two or more line numbers are specified, then those lines are deleted along with any lines which lie between the

specified lines. The second line number specified must be greater than the first. If any lines are entered underneath the $C command, they will be inserted between the line before and the line after the group of lines which was deleted. The insertion lines end when a command keyword is entered.

<u>$L ( ) ( )</u> - List lines or whole block. This command causes one or more lines to be listed along with the corresponding line numbers. When no line numbers are included in the command, the entire block is listed. If a single line number is included, then corresponding line and all subsequent lines in the block are listed. When two line numbers are entered with the command, the two corresponding lines and any intervening lines are listed. The second line number specified must be greater than or equal to the first. When both line numbers specified are the same, then a single line is listed. If a listing is specified after changes or insertions have been made but before a renumber operation ($N), then the inserted lines will be listed without the line numbers. The line numbers displayed in a listing are valid until a renumber is performed on the block.

<u>$N</u> - Renumber the block. This command causes new line numbers to be assigned to each line so that every line is numbered. Lines inserted under the '$A' or '$C' commands will be unnumbered until a $N command is used. The $N command also removes any restriction on the line numbers which can be addressed by the $C and $A commands. The $C and $A commands must always address lines sequentially until an intervening $N command is used. For example, if '$A 30' is entered, then '$C 10 20' is an illegal command until a '$N' command is used.

<u>$E</u> - End change mode. This command causes the change mode to exit. When '$E' is entered, the following message is printed.

ENTER STORE OR KILL

The user should enter 'STORE' or 'KILL' as appropriate. If 'STORE' is entered, the block specified by name when the 'EDIT/CHANGE' mode was entered is replaced by the modified block just created. If 'KILL' is entered, the specified block is unchanged and the modifications specified under the 'EDIT/CHANGE' mode are discarded. After 'STORE' or 'KILL' is entered, the computer returns to normal user command step input mode.

$? - List available commands. This command causes the computer to list all six avaliable commands along with brief descriptions of each. The same listing is provided automatically when the 'EDIT/CHANGE' mode is first entered.

A command keyword for the EDIT/CHANGE mode must be the first entry on a line. The following two entries may be line numbers as needed. Two entries, keyword or numeric, should be separated by a comma or one or more blanks. Only one command keyword may appear on a line. Lines for insertion should not be included on the same line as command keywords. The inserted lines will be copied exactly as entered, without error checking.

5.7.2 Tektronix Cross-Hair Cursor

When a Disposition substep specifies that the Tektronix Graphics Cursor shall be activated, the cursor cross-hairs appear on the screen following completion of the plot. The user can manipulate the graphic cursor by using the knobs on the right-hand side of the Tektronix keyboard. When the cross-hair intersection indicates a point on the plot which the user wishes evaluated, any key on the keyboard can be struck to extract the coordinates of the point in the scale of the plot. The coordinate values are printed on the left-hand side of the screen.

The coordinate values printed represent the position of the cross-hair cursor on the plot. The user must assure that the cursor intersection is actually located on the plot point which is to be evaluated. The Tektronix screen actually consists of a raster matrix of points which may be addressed. The raster separation limits the accuracy of both the plot itself and the graphic cursor evaluation of points on the plot. The computer indicates this potential inaccuracy by calculating a number equivalent to one-half the raster separation in the scale of the plot. This number is printed for

both the dependent and independent variables just to the right of the corresponding scale value. A typical Graphic Cursor point evaluation is printed:

```
0.9054E-01 +/- .94E-03      0.1476E+02 +/- .94E-03
```

The left-hand values are the independent variable location and accuracy, respectively, and the right-hand values are the dependent variable location and accuracy.

If the user enters the character 'C' to request a point evaluation, then the Graphics Cursor will remain active after the location has been printed so that the cursor may be repositioned for additional point evaluations. When any character except 'C' is entered, then the Graphics Cursor is deactivated after the current location is evaluated and the computer will proceed to the next command step.

Point locations may be evaluated from log-log or semilog plot scales as well as from linear scales. The user may notice that the computed accuracy of the point evaluation changes dramatically as cursor position is changed on a log scale.

5.7.3 Printout Control

When printed output is generated in the Interactive or Interactive Graphics mode, the printout is written a page at a time. At the end of each page, the user is informed of the independent variable interval included on the next page and given the option to print the next page, skip the next page, or skip the balance of the printout for the current row and column position. An example of the message which gives this information and option is:

```
NEXT PAGE .9375E+00 TO .9961E+00 ENTER:C=CONTINUE,S=SKIP,X=STOP
```

This message was compressed slightly to fit on this page.

If the user enters 'C', the next page will be printed and, if the printout is incomplete, the above message will be repeated with the interval changed to correspond to the subsequent page. If the user enters 'S', the next page will not be printed and if the printout is incomplete, the above message will be repeated for the subsequent page. If the user enters 'X', the balance of the values for the current row and column position are skipped and printout begins for the next row/column position. If the last row and column are currently being printed, or if there is a single data stream being printed, then the computer proceeds to the next command step.

5.7.4 Menu Listing Control

Menu listing control is very similar to printout control. When a menu of item codes or counters is generated, the menu is printed in 50-line blocks. Frequently, all of the item codes or counters will fit in one block. However, if multiple blocks are required, then a message similar to the following example will appear at the end of each block.

NEXT BLOCK P985 TO T004 ENTER: C=CONT, S=SKIP, X=STOP

The meaning of these control entries is the same as the print-out control entries except that the entry 'X' will always cause the computer to proceed to the next command step.

5.8 INFO FILE FORMAT

The 'Info' file is provided as a method to simplify user specification of large numbers of item codes representing like sensors, geometric positions of sensors, and labeling information for output of processes using data from these sensors. The file can be maintained in the same way that program source decks and formatted data decks are maintained. For the OLS application, two specific Info files are provided. However, the user is free to create a new Info file or modify the existing OLS Info file for a special application. Naturally, the user must stay within the specified format for the Info file and, for certain derivations, must follow processing conventions and provide all required information for the processes to be performed.

The Info file consists of one or more groups. Each group except the initial group has a four-character name, and the last line of every group contains only the keyword 'END'. The final line of the Info file must be the keyword '$$$$' located in the first four character positions.

The rules for specifying strings and numbers in the Info file are the same rules used in generating command steps. A comma or a number of blanks may be used as a separator and the slash is a special separator. However, no default entries are allowed and all four letters of every keyword must be included.

The format for the initial group is different from the format for all the other groups. The initial group must always be present at the top of the Info file. This group consists of a number of keyword pointers along with corresponding item codes and some numeric inputs. The keywords are Processing Program

identification pointers for data needed for certain derivations. Each keyword corresponds to some measured parameter which will be required by the program. Following is a list of keywords the program will recognize and the corresponding measured parameters:

MRAZ - Main Rotor Azimuth
TRAZ - Tail Rotor Azimuth
TIAS - True Indicated Airspeed (Knots Squared)
OATM - Outside Air Temperature (Deg C)
STAT - Static Pressure (PSIA)
MTOR - Main Mast Torque (in.-lb)
TTOR - Tail Mast Torque (in.-lb)

Each keyword listed in the initial group begins a sequence of entries with each sequence terminated by a slash. The subsequent entries in the sequence include one or more item codes and, following each item code, zero or one number. Each item code corresponds to the type of measured parameter specified by the keyword. The number which may follow an item code is supplied as input to some derivation which uses the corresponding measured parameter. For example, each OLS rotor azimuth item code is followed by a number which specifies an offset in degrees for correction of the azimuth values. Refer to the top of Figure 25 for an example of the initial group of an Info file.

Each subsequent group in the Info file contains a one- or two-dimensional array of item codes or pairs of item codes. For two-dimensional arrays, each row and column element has a geometric location. One-dimensional arrays have geometric locations only for the column elements. The first two characters for a two-dimensional group must be 'S2', while the corresponding two characters for a one-dimensional group must be 'S1'.

A group title follows the group name on the same line. The group title is separated from the group name by at least one blank. The title begins with the first nonblank character following the group name and may include as many as 30 characters. This title is used for the menu display of the Info file and as a plot title when multiple item codes are accessed using the Info file.

The next three lines include a column position scale label, a shortened column position scale label, and a geographic feature. The column position scale label may include as many as 16 characters. This label is a title for the column scale values which are listed later in the group. The shortened

```
MRAZ R992 0.0, R106 0.0, R018 30.32/
TRAZ R025 45.0/
TIAS P002/
OATM T004/
STAT P030/
MTOR M107/
END
S1BV  BLADE BEAMWISE VIBRATION
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.2273,.3087,.3902,.5000,.5902,.7000,.9020,.9962//
BLBV//
A938/A939/A940/A950/A951/A952/A953/A954//
END
S1CV  BLADE CHORDWISE VIBRATION
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.2273,.3087,.3902,.5000,.5902,.7000,.9020,.9962//
BLCV//
A955/A956/A967/A968/A969/A970/A971/A972//
END
S2BU  BL BUTTONS   UPPER SURFACE
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.40,.60,.75,.864,.955//
FRACTN OF CHORD
X/CHORD
LEADING EDGE
.30,.60,.90//
BLUI,BLUO//
P758,.999,P759,-.985/P770,.862,P771,-.916/
P750,-.934,P751,.949/P732,.893,P733,-.997/
P982,-.949,P983,.922//
P760,-.886,P761,.992/P772,.945,P773,-.976/
P752,-.934,P753,1.007/P734,.937,P735,.925/
P984,.971,P985,-1.004//
P762,-.881,P763,-1.058/P741,-.940,P742,.869/
P754,.768,P725,-.825/P736,-.905,P737,.946/
P986,-.992,P987,1.004//
END
S2BL  BL BUTTONS   LOWER SURFACE
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.40,.60,.75,.864,.955//
FRACTN OF CHORD
X/CHORD
LEADING EDGE
.30,.60,.90//
BLLI,BLLO//
P764,-.966,P765,.982/P743,.776,P744,-.799/
P726,-.998,P727,-1.109/P976,-.941,P977,-1.006/
P988,-.980,P964,1.036//
P766,-.877,P767,.921/P745,.772,P746,-.939/
P728,-.918,P729,.865/P978,.980,P979,-1.025/
P965,-.944,P966,.959//
P768,.890,P769,-.845/P747,.825,P748,-.935/
P730,-.943,P731,.976/P980,.990,P981,-.983/
P755,-.941,P756,.967//
END
```

Figure 25. OLS Info File.

```
S2HW   HOT-WIRE ATTENUATION SENSORS
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.4000,.5924,.7500,.8639,.9545//
CONTOUR POSITION
INCHES
LEADING EDGE
-1.56,-1.44,-1.32,-1.20,-1.08,-0.96,-0.84,-0.72,
-0.60,-0.48,-0.36,-0.24,-0.12,0.0,0.12,0.24,0.36//
HWAT//
NULL/V171/V820/V866/V923//NULL/V172/V821/V867/V924//
NULL/V183/V832/V878/V925//NULL/V184/V833/V879/V929//
NULL/V185/V834/V880/V930//NULL/V186/V835/V881/V931//
NULL/V800/V846/V894/V932//V151/V801/V847/V895/V933//
V152/V802/V848/V896/V934//V153/V803/V849/V897/V944//
V154/V804/V850/V898/V945//V155/V805/V851/V899/V946//
V156/V816/V862/V911/V947//V167/V817/V863/V913/V949//
V168/V818/V864/V915/V961//V169/V819/V865/V912/V963//
V170/NULL/NULL/NULL/NULL//
END
S1BB   BLADE BEAMWISE BENDING
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.0227,.2273,.3087,.3902,.5000,.5902,.7000,.8042,.9020//
BLBB//
B112/B120/B126/B128/B122/B130/B132/B124/B134//
END
S1CB   BLADE CHORDWISE BENDING
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.0227,.0436,.2273,.3087,.3902,.5000,.5902,.7000,.8042,.9020//
BLCB//
B113/B115/B121/B127/B129/B123/B131/B133/B125/B135//
END
S1BT   BLADE TORSION
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.0227,.3087,.5000,.7000,.9020//
BLTR//
M906/M150/M935/M936/M937//
END
S2PP   BLADE ABSOLUTE PRESSURE
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.40,.60,.75,.864,.955//
FRACTN OF CHORD
X/CHORD
LEADING EDGE
.009991,.029972,.079930,.149829,.199825,.249782,.349694,
.399651,.449607,.499563,.549520,.599476,.699389,.919196//
BLAP,BLAM//
```

Figure 25. (Continued).

```
P157,.016697,P173,-.016697/P187,.016697,P809,-.016697/
P828,.016697,P856,-.016697/P184,.016697,P831,-.016697/
P908,.016697,P958,-.016697//
P158,.026953,P174,-.026953/P188,.026953,P810,-.026953/
P836,.026953,P857,-.026953/P185,.026953,P843,-.026953/
P909,.026953,P959,-.026953//
P159,.039120,P175,-.039120/P189,.039120,P811,-.039120/
P837,.039120,P858,-.039120/P186,.039120,P844,-.039120/
P919,.039120,P973,-.039120//
NULL,.046362,NULL,-.046362/P190,.046362,P812,-.046362/
P838,.046362,P868,-.046362/P180,.046362,P845,-.046362/
P920,.046362,P974,-.046362//
NULL,.048165,NULL,-.048165/NULL,.048165,NULL,-.048165/
P839,.048165,P869,-.048165/P181,.048165,P859,-.048165/
NULL,.048165,NULL,-.048165//
P160,.048164,P176,-.048164/P191,.048164,P822,-.048164/
P840,.048164,P870,-.048164/P182,.048164,P860,-.048164/
P921,.048164,P975,-.048164//
NULL,.044446,NULL,-.044446/P192,.044446,P823,-.044446/
P841,.044446,P871,-.044446/P194,.044446,P861,-.044446/
P926,.044446,P989,-.044446//
NULL,.041355,NULL,-.041355/NULL,.041355,NULL,-.041355/
P842,.041355,P872,-.041355/P195,.041355,P875,-.041355/
P927,.041355,P990,-.041355//
P161,.038071,P177,-.038071/P193,.038071,P824,-.038071/
P852,.038071,P873,-.038071/P196,.038071,P876,-.038071/
P928,.038071,P991,-.038071//
NULL,.034788,NULL,-.034788/NULL,.034788,NULL,-.034788/
NULL,.034788,NULL,-.034788/P813,.034788,P877,-.034788/
P941,.034788,P738,-.034788//
NULL,.031504,NULL,-.031504/P806,.031504,P825,-.031504/
P853,.031504,P874,-.031504/P814,.031504,P891,-.031504/
P942,.031504,P739,-.031504//
NULL,.028220,NULL,-.028220/NULL,.028220,NULL,-.028220/
NULL,.028220,NULL,-.028220/P815,.028220,P892,-.028220/
NULL,.028220,NULL,-.028220//
P162,.021653,P178,-.021653/P807,.021653,P826,-.021653/
P854,.021653,P884,-.021653/P829,.021653,P893,-.021653/
P943,.021653,P740,-.021653//
P163,.007205,P179,-.007205/P808,.007205,P827,-.007205/
P855,.007205,P885,-.007205/P830,.007205,P907,-.007205/
P957,.007205,P757,-.007205//
END
$$$$
```

Figure 25. (Concluded).

column position scale label has the same meaning as the previous line but is limited to eight characters. The geographic feature is a label for some structural reference point that is closest to the first column element. In the OLS Info file, this feature is invariably 'BLADE ROOT'. This label may include as many as 16 characters.

The next line gives the geometric position for the column elements in ascending order of element numbers. This 'line' may, in fact, extend over one or more lines of input. The column position list is terminated by two slashes. Individual column positions are separated by one or more blanks or a comma. The number of column positions entered in this list specifies the number of column elements in the item code array.

Two-dimensional groups now have four lines of additional input which are the same in meaning as the last four except that the row elements and scale are described. One-dimensional groups omit these four lines.

Following row and column labels and element positions, there must be a line of one or two keywords corresponding to the double-row elements. This keyword list is terminated by two slashes. The number of keywords specified in the line corresponds to the number of double-row elements in the group. If two keywords are present, the first keyword must correspond to the top double-row element and the second keyword to the bottom.

Following the keyword line is the list of item codes. For each row position, there is a sequence of item code entries, which is terminated by a double slash. This sequence may extend over one or more lines. Inside each sequence, the information corresponding to each column position is delimited by a single slash to form a matrix element entry. The sequence of matrix element entries must correspond to the sequence of column positions.

Each matrix element entry must contain one item code for each double-row element. The first item code in the sequence must correspond to the top double-row element. If there is no item code (i.e., no sensor) for a particular double-row element of a row/column position, the entry 'NULL' should be used. Immediately following each item code, a numeric value may be entered. Such a value might be required for a derivation process. For example, the C_n integration requires the vertical as well as horizontal position of a sensor for a chord profile. An example of a matrix element entry is the row one, column one array position entry for the group 'S2PP'

P157, .016697,P173,-.016697/

as shown in Figure 25.

Each sequence of matrix element entries which form a row must begin on a new line. Each row must contain the same number of matrix elements as the specified number of column positions. For a two-dimensional matrix, the order of the rows must correspond to the order of the row position list and there must be the same number of rows as row positions. A one-dimensional group must contain a single row.

The standard convention for the OLS blade application of this program has been that column positions in the Info file correspond to blade radial stations and row positions correspond to chord stations. The blade static pressure, blade normal force coefficient, blade chordwise force coefficient, blade pitching moment coefficient, blade displacement, and blade slope derivations all presume that this convention has been followed. However, the user is free to change the convention for other applications.

As mentioned in Section 4.7, two Info files are provided for the OLS blade application. The sample Info file shown in Figure 25 is for the Red blade. For the White blade, the second Info file (not shown) contains only the initial group and the group 'S2PP' which specifies the blade absolute pressure sensors. The azimuth correction offsets for this Info file are adjusted by 180 degrees to align White blade azimuth with the measured azimuth blip which nominally occurs when the Red blade is over the tail boom.

6. PROCESSING ALGORITHMS

The purpose of this section is to acquaint the user with the mathematical techniques used by the Processing Program ANALYZE or DERIVE command steps. Methods which are well defined and documented in the mathematical literature are covered with brief descriptions and references to that literature. The specific nature of application of these methods to the data streams addressed by this program will be covered in full in this section.

6.1 ANALYSES

6.1.1 Harmonic Analysis

Numerous texts show that a very broad class of continuous and piecewise continuous functions, x(t), may be represented by infinite series of trigonometric sine and cosine functions over a finite interval, T, as

$$x(t) = \frac{a_0}{2} + \sum_{K=1}^{\infty} \left(a_K \cos \frac{2\pi Kt}{T} + b_K \sin \frac{2\pi Kt}{T}\right), \quad (1)$$

where

$$a_K = \frac{2}{T} \int_0^T x(t) \cos \frac{2\pi Kt}{T}\, dt$$

and $K=0,1,2,\ldots$

$$b_K = \frac{2}{T} \int_0^T x(t) \sin \frac{2\pi Kt}{T}\, dt$$

The series is referred to as a Fourier Series and the coefficients, a_K and b_K, are referred to as Fourier coefficients.

Oftentimes, the series is expressed in a different form as derived below:

$$x(t) = \frac{a_0}{2} + \sum_{K=1}^{\infty} \sqrt{a_K{}^2+b_K{}^2} \left[\frac{a_K}{\sqrt{a_K{}^2+b_K{}^2}} \cos \frac{2\pi Kt}{T} + \right.$$

$$\left. \frac{b_K}{\sqrt{a_K{}^2+b_K{}^2}} \sin \frac{2\pi Kt}{T} \right]$$

If a_K and b_K are not both zero, there exists a number, ϕ_K, such that

$$\cos \phi_K = \frac{a_K}{\sqrt{a_K{}^2+b_K{}^2}} \text{ and } \sin \phi_K = \frac{b_K}{\sqrt{a_K{}^2+b_K{}^2}}$$

After defining

$$c_K = \sqrt{a_K{}^2+b_K{}^2}$$

we may write

$$x(t) = \frac{a_0}{2} + \sum_{K=1}^{\infty} c_K \left[\cos \phi_K \cos \frac{2\pi Kt}{T} + \sin \phi_K \sin \frac{2\pi Kt}{T}\right]$$

Using the well known trigonometric identity,

$$\cos (\alpha-\beta) = \cos \alpha \cos \beta + \sin \alpha \sin \beta$$

the series may be written as

$$x(t) = \frac{a_0}{2} + \sum_{K=1}^{\infty} c_K \cos \left(\frac{2\pi Kt}{T} - \phi_K\right) , \tag{2}$$

where

$$\phi_K = \tan^{-1} \left(\frac{b_K}{a_K}\right)$$

Thus, referring to Equation (2), each term in the series is called a harmonic and the harmonic number is K. In addition, the frequency in hertz of each term is given by

$$\frac{K}{T} ,$$

while the magnitude or amplitude of each term is given by c_K and the phase is given by ϕ_K.

In practical applications involving signal processing, the data are not known as a continuous function nor is it possible to compute an infinite number of harmonics. Fortunately, only a few harmonics are of interest in many applications. Text books on numerical analysis show that the function, x(t),

may be approximated by finite trigonometric series using a sequence of values of the function at various times, t. In particular, let

$$x_j = x(t_j),$$

where

$$t_j = j\Delta t, \; j=0,1,2\ldots,N-1.$$

Then it can be shown that for M sufficiently large,

$$x_j \cong \frac{a_0}{2} + \sum_{K=1}^{M} \left(a_K \cos \frac{2\pi Kj}{N} + b_K \sin \frac{2\pi Kj}{N}\right), \qquad j=0,1,\ldots,N-1 \tag{3}$$

where

$$a_K = \frac{2}{N} \sum_{j=0}^{N-1} x_j \cos \frac{2\pi Kj}{N}$$

$$b_K = \frac{2}{N} \sum_{j=0}^{N-1} x_j \sin \frac{2\pi Kj}{N} \qquad K=0,1,2,\ldots,M$$

and $M \leq N$

Thus, Equation (3) is analogous to Equation (1). It can also be shown that the trigonometric series given by Equation (3) is the best approximation in the least squares sense to the sequence of values, x_j, over all other trigonometric (sine and cosine) series having no more than M terms. If the number of terms, M, is chosen to be N, the series given by Equation (3) is identically x_j and is called the Discrete Fourier Transform.

The Harmonic Analysis algorithm requires as input a time history with a length and start point corresponding to an integer number of rotor cycles. Fourier components are only calculated for those Harmonic numbers which correspond to a positive integer multiple of the number of cycles represented in the input data. The output harmonics of this process are scaled as

the sequential harmonic components of a single rotor cycle of data. When more than one rotor cycle of input data are provided, this process provides a more accurate estimate of steady-state amplitude. In addition, inaccuracies in rotor azimuth are minimized by processing more than a single rotor cycle.

The computational routine is a recursive algorithm described in Reference 2. This process presumes that frequencies of interest in the data are integral multiples of the rotor speed expressed in revolutions per second.

For output, the sine and cosine harmonic terms are converted to amplitude and phase in degrees. The output independent variable scale is normally displayed as Harmonic number for a single rotor cycle. However, the user has the option to specify frequency as the independent variable. Frequency may only be specified as the independent variable in the same command step in which the Harmonic analysis is actually calculated.

6.1.2 Digital Filtering

Digital filtering operations in the OLS/DMS are accomplished using Chebyshev filters. The magnitude characteristic of these filters varies between equal maximum and equal minimum values in the passband and decreases monotonically toward zero outside this frequency band. In particular, the square of the magnitude of the Chebyshev transfer function is defined by

$$H(i\omega)^2 = \frac{1}{1+\delta^2 T_N^2(\omega)},$$

where $T_N(\omega)$ is the Nth degree Chebyshev polynomial which may be defined by

$$T_N(1) = \begin{cases} \mathrm{Cos}\,(N \cos^{-1}\omega), & 0\leq\omega\leq 1 \\ \mathrm{Cosh}\,(N \cosh^{-1}\omega), & \omega>1. \end{cases}$$

---

[2] A. Ralston and H. Wilf, MATHEMATICAL METHODS FOR DIGITAL COMPUTERS, John Wiley and Sons, New York, 1960, Chapter 24.

The constant δ is the amplitude of the oscillation in the passband, $i = -1$, ω is in units of angular frequency normalized to unity, and N is the number of poles in the transfer function. In general, the phase characteristics of the Chebyshev filters are undesirable. However, a procedure involving two filtering operations on the data results in an effective filter which is phase free and distortionless. A mathematical analysis of the technique is given in Reference 3. To illustrate the technique and to also show the magnitude/phase characteristics of a seven-pole Chebyshev filter, a swept frequency sinusoidal function was generated and filtered. Following the procedure, the filter output was manipulated then filtered again. Using Fast Fourier Transform techniques, together with system input/ output relations, the frequency response function (transfer function) was computed and is displayed in Figure 26. The user is provided the flexibility of selecting both low-pass and band-pass filters together with the number of poles to be used.

The decrease in magnitude of the filter transfer function with increased frequency above the filter breakpoint is called the <u>rolloff</u> of the filter. Examination of the formulas for the transfer function and the Chebyshev polynomial shows that the rolloff is more rapid for large N and less rapid for small N. The speed with which the transfer function magnitude decreases with increased frequency above the filter breakpoint is called the <u>rate of rolloff</u>.

Figures 27 and 28 show main rotor shaft horsepower for various airspeeds before and after the application of a low-pass digital filter. The filter was specified to have four poles and a break frequency of 2.5 Hz.

Filtering processes tend to distort finite records near the boundaries of these records. The filtering algorithm seeks to minimize this distortion by synthesizing extensions to the beginning and end of the input data record. Every synthesized data value at the beginning of the data record is set equal to the first real data value. Similarly, every synthesized data value for the end of the data record is set equal to the last real data value. These artificial extensions to the input data record are not retained on output.

---

[3] A. L. Eubanks, FILTER DESIGN AND ANALYSIS WITH APPLICATIONS TO DISCRETE DATA, Bell Helicopter Textron Report 299-099-889, Fort Worth, Texas, 15 August 1977.

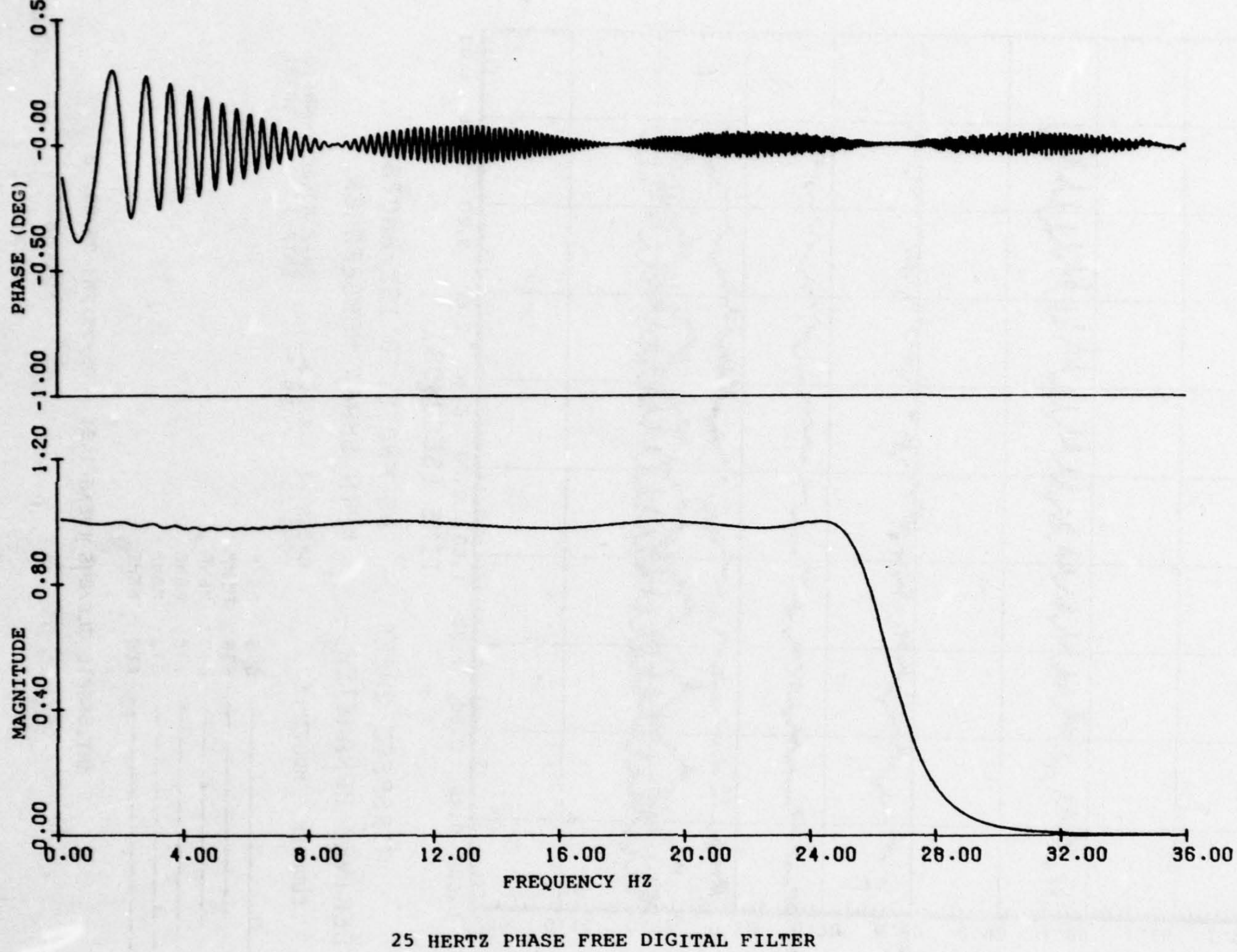


Figure 26. Two pass Chebyshev filter characteristics.

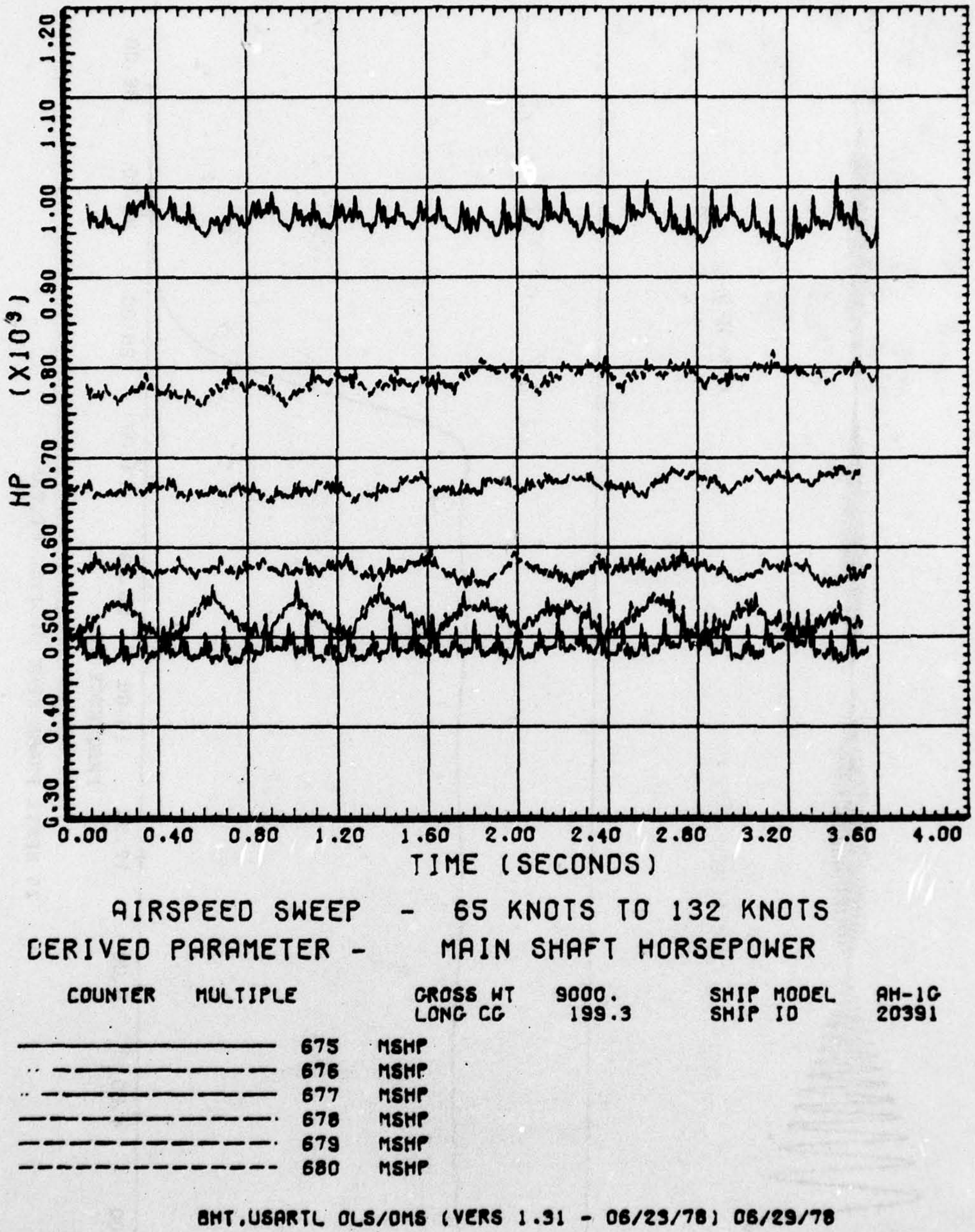


Figure 27. Main rotor shaft horsepower for several maneuvers before digital filtering.

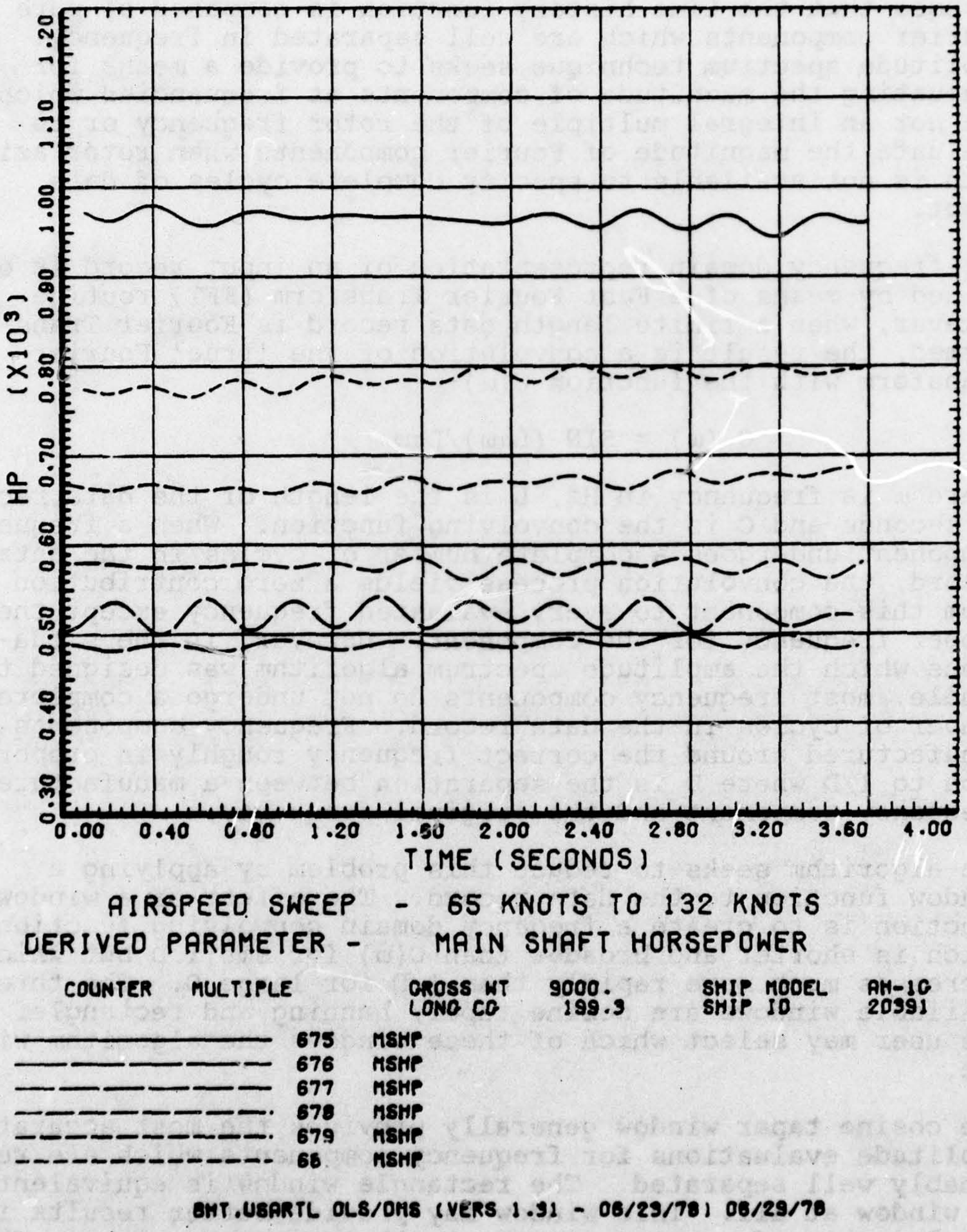


Figure 28. Main rotor shaft horsepower for several maneuvers after digital filtering (2.5 Hz, 4 poles).

6.1.3 Amplitude Spectra

The amplitude spectrum routine evaluates the amplitude of the Fourier components of an input time history. The algorithm assumes that the time history function is composed of pure Fourier components which are well separated in frequency. The amplitude spectrum technique seeks to provide a means for evaluating the magnitude of components at frequencies which are not an integral multiple of the rotor frequency or to evaluate the magnitude of Fourier components when rotor azimuth is not available to specify complete cycles of data input.

The frequency domain representation of an input record is obtained by means of a Fast Fourier Transform (FFT) routine. However, when a finite length data record is Fourier Transformed, the result is a convolution of the 'true' Fourier Transform with the function $C(\omega)$

$$C\ (\omega) = \mathrm{SIN}\ (L\pi\omega)/L\pi\omega$$

where ω is frequency in Hz, L is the length of the data record in seconds and C is the convolving function. When a frequency component undergoes a complete number of cycles in the data record, the convolution process yields a zero contribution from this component to every evaluated frequency except the proper frequency for the component. However, in the situations which the amplitude spectrum algorithm was designed to handle, most frequency components do not undergo a complete number of cycles in the data record. Frequency components are manufactured around the correct frequency roughly in proportion to 1/D where D is the separation between a manufactured frequency component and the original frequency.

The algorithm seeks to reduce this problem by applying a window function to the data record. The effect of a window function is to create a freqency domain convolving function which is shorter and broader than $C(\omega)$ for small D but which decreases much more rapidly than 1/D for large D. The three available windows are cosine taper, hanning and rectangle. The user may select which of these windows the algorithm will use.

The cosine taper window generally provides the most accurate amplitude evaluations for frequency components which are reasonably well separated. The rectangle window is equivalent to no window at all. This window may provide better results in resolving two frequency components that are very close. The

hanning window should be used when adjacent frequency components are well separated but several orders of magnitude different in amplitude.

After the window function is applied to the data record, the FFT is performed. The number of points in the input data record is constrained to be a power of two so as to optimize the computational efficiency of the FFT algorithm. After the FFT is performed, the magnitude squared for each frequency component is computed and then a corrective factor is applied to compensate for the effects of the window function. After the corrective factor is applied, the algorithm computes the sums for each set of three adjacent squared magnitude frequency components. Next, the square root of each resultant squared magnitude sum is computed. The result of this process is an amplitude spectrum estimate for which, given the initial assumptions for the algorithm, the peak values displayed are accurate estimates of the amplitude of pure frequency components.

6.1.4 Moving Block Damping Estimation

The Moving Block Damping Analysis algorithm assumes that the input function, or time history, is of the form

$$f(t) = Ae^{-\frac{D\omega t}{100}} \sin(\omega t+Q) + Q(t),$$

where D is percent of critical damping, ω is a known or suspected frequency component, and $Q(t)$ is a function with frequency components well separated from ω. The algorithm extracts a single value, the percent of critical damping D, for each time history processed.

From the input time history, a sequence of overlapping blocks of data is chosen for analysis such that each block is identical to the preceding block except that the next data sample is included in the block and the first data sample of the preceding block is excluded. Then a Fourier analysis is performed on each block and the logarithm of each magnitude at frequency ω is computed to yield a curve which has a slope equal to the damping associated with the frequency, ω. The principle of least squares is applie to the sequence of log values to

obtain the best estimate of damping. Reference 4 reports on the use of the method.

6.1.5 Cycle Averaging

The cycle averaging algorithm seeks to reduce superfluous noise by averaging together several contiguous cycles to form a single representative cycle. Initially, 256 equally spaced azimuth positions are established. From each input cycle of data, the computer interpolates values for every established azimuth position. Then the values corresponding to each azimuth position are averaged together.

6.1.6 Min/Max Analysis

Data are evaluated in the sense of min/max through the following process. Input data consisting of an integer number of complete cycles are examined, cycle by cycle, for the minimum and maximum value occurring in each cycle. Then, for every cycle, mean and oscillatory values are computed by

$$\text{Osc} = 1/2\ (\text{max} - \text{min})$$
$$\text{mean} = 1/2\ (\text{max} + \text{min})$$

6.2 DERIVATIONS

6.2.1 Rotor Azimuth

Although rotor azimuth is a measured parameter, the rotor azimuth data stream must be processed to yield useable azimuth angles. The input data stream for rotor azimuth consists of a sequence of negative values broken by either a single positive value or two adjacent positive values which occur nominally at an azimuth angle of zero degree (red blade over the tail boom). A time instant is found for each positive value or adjacent pair of positive values. The correction offset angle supplied by the Info file is then applied by interpolating between adjacent nominal zero degree time instants to get corrected zero degree time instants. Jitter in these time instants is minimized by averaging each set of three adjacent zero degree time instants. Azimuth angle associated with each data value is then computed from estimated time instants of 360K degrees (K an integer) using linear interpolation.

---

[4] J. G. Yen, S. Viswanathan, and C. G. Matthys, FLIGHT FLUTTER TESTING OF ROTARY WING AIRCRAFT USING A CONTROL SYSTEM OSCILLATION TECHNIQUE, NASA Symposium on Flutter Testing Techniques, Flight Research Center, Edwards AFB, Calif, October 9-10, 1975.

6.2.2 Vehicle True Airspeeds

Since indicated airspeed was measured and recorded on the Operational Load Survey data tapes, several computations are required to determine vehicle true airspeed. The parameters, Outside Air Temperature (°C), Boom System Static Pressure (psia), and Boom System Airspeed (knots squared), are recorded on tape and are used in the computation.

First, the Boom System Airspeed in knots squared is smoothed by averaging together all values from three adjacent rotor cycles, and one smoothed value is produced for each rotor cycle. Then, from the smoothed Boom System Airspeed, $v_i^2(t)$, calibrated airspeed is determined according to

$$v_C(t) = m\, v_i\,(t) + b,$$

where the constants m and b are not the calibration constants on tape and depend on the particular gage that was used. Vehicle true airspeed, $v_T(t)$, is then computed from

$$V_T(t) = \frac{v_C(t)}{\sqrt{\sigma(t)}},\ \text{knots}$$

where

$$\sigma(t) = \frac{\rho(t)}{\rho_o},$$

ρ_o is the sea level standard air density (.002378 slugs/ft^3) and

$$\rho(t) = \frac{\rho_o\, T_o\, P_s(t)}{P_o\, T_A(t)},\ \frac{\text{slug}}{\text{ft}^3}\ .$$

$P_s(t)$ is the Boom System Static Pressure, T_o is the sea level standard temperature (518.7°R), P_o is the sea level standard pressure (14.7 psi), and $T_A(t)$ is the outside air temperature in °R, which is computed from measurements by

$$T_A(t) = 1.8\ T(t) + 491.7,$$

where T(t) is the measured outside air temperature in °C.

6.2.3 Rotor RPM

Rotor speed is evaluated for each zero degree azimuth instant by averaging the cycle intervals before and after that instant, taking the reciprocal of the average, and converting the resultant rotor frequency to RPM.

In order to obtain a function (i.e., RPM) with sample rate identical to other parameters of interest, a simplified cubic spline interpolation procedure is employed. The method seeks to minimize large oscillations between points and has the advantage that continuity of slope is achieved at measured data values. The method generally achieves smooth curves and is computationally efficient.

6.2.4 Rotor Shaft Horsepower

Main or tail rotor shaft horsepower is computed from the measured parameters, main rotor mast torque (in.-lb) and tail rotor mast torque (in.-lb), making use of the derived parameters, main rotor RPM, and tail rotor RPM. Shaft horsepower for either main or tail rotor is then given by

$$HP(t) = K_0 K_1 K_2 \; Q(t) \; R_{PM}(t) \text{ , where}$$

K_0 is the conversion factor $2\pi/60$ converting the appropriate RPM to radians/sec, K_1 is the conversion factor 1 horsepower/550 $\frac{\text{ft-lb}}{\text{sec}}$, K_2 is the conversion from inches to feet, 1/12, $Q(t)$ is the appropriate mast torque, and $R_{PM}(t)$ is rotor RPM.

6.2.5 Thrust Coefficient

The thrust coefficient, C_T, is computed from Boom System static pressure (psia), outside air temperature (°C), and rotor RPM. C_T is given by

$$C_T = \frac{W}{\sigma \rho_o A (\Omega R)^2} \text{ ,}$$

where W is the ship gross weight, σ is the density ratio as described in Paragraph 6.2.2, ρ_o is sea level standard air density, A is area of the rotor disc (πR^2), and ΩR is the rotor tip speed

$$\Omega R = \frac{R_{PM}(t)R}{K}\text{, where}$$

$R_{PM}(t)$ is rotor RPM, R is rotor radius, and K is the conversion from RPM to radians/sec (60/2).

6.2.6 Torque Coefficient

The torque coefficient, C_Q, is computed from measurements of rotor mast torque (in.-lb), Boom System static pressure (psia), outside air temperature (°C), and the derived rotor RPM. In particular,

$$C_Q = \frac{Q(t)\ R_{PM}(t)\ K_0 K_1}{\sigma\rho_o\ A(\Omega)^3}\text{, where}$$

Q(t) is rotor mast torque, $R_{PM}(t)$ is rotor RPM, K_0 is conversion from RPM to rad/sec (2 /60), K_1 is conversion from inches to feet (1/12), σ is air density ratio described in Paragraph 6.2.2, ρ_o is sea level standard air density (.002378 slugs/ft^3), A is the area of the rotor disc (R^2), and ΩR is the rotor tip speed described in Paragraph 6.2.5.

6.2.7 Blade Local Flow Magnitude and Direction

Blade local flow is derived from the two perpendicular measurements of differential pressure, q_1 and q_2, recorded from a single boundary layer button (BLB). In addition, boom system static pressure and outside air temperature measurements are used.

As described in reference 1, a BLB consists of two total pressure tubes and a static port. These two tubes are mounted at an angle of 90 degrees to each other. For the OLS measurement program, the bisector of this right angle was always aligned with the blade chordline. However, in case the BLBs are rotated for another measurement program, the Processing Program allows the user to enter the angle, B, between the inboard pointing tube and the chordline. The default value for B is 45 degrees. The differential pressure measured using the inboard pointing tube and the static port is called $q_1(t)$ and the other differential pressure measurement is called $q_2(t)$.

The BLB flow angle, A_B, is the angle made by the flow and the bisector of the angle between the tubes. Negative $A_B(t)$ values imply flow more nearly aligned with the inboard pointing tube than the outboard pointing tube. $A_B(t)$ is computed by reference to calibration values which relate the flow angle in degrees and the ratio of the two differential pressures. The Info File format accommodates two calibration constants, C_1 and C_2, for each BLB (one number for each item code). The computational algorithm uses these two numbers as corrective multipliers where C_1 follows the inboard pointing item code in the Info File (the first item code listed for each sensor is the inboard) and C_2 follows the outboard pointing item code. C1 applies to inboard or negative angles and C2, to outboard or positive angles.

The calculation of BLB flow angle is made by one of two formulas

$$A_B(t) = C_1 \; F_N(q_2(t)/q_1(t))$$

if $q_2(t)$ is less than $q_1(t)$ (negative angle) or

$$A_B(t) = C_2 \; F_N(q_1(t)/q_2(t))$$

if $q_1(t)$ less than $q_2(t)$ (positive angle). F_N is one of two polynomial functions

$$F_1(R) = ((1.3R-11.3)\;R + 25.6)\;R - 15.6$$

$$F_2(R) = ((.059233R + 8.732)\;R - 26.23)\;R + 17.437$$

where $R = q_1/q_2$ or $R = q_2/q_1$. F_1 and F_2 are rough functional approximations of the BLB calibrations but are insufficiently accurate for application to individual BLB's without the correction factors C_1 or C_2.

F_1 is selected for negative angles when C_1 is positive and for positive angles when C_2 is negative. F_2 is selected for positive angles when C2 is positive and for negative angles when C_1 is negative. Thus the 'fit' to the calibration points is made by selection of one of two third degree functions and a correcting multiplicative factor based first on the relative size of q_1 and q_2 and then on the sign of the correction factor.

Figures 29 and 30 show representative comparisons of corrected polynomials (continuous lines) and calibration points (squares) for two BLBs. Similar corrected polynomial plots are available for every BLB used in the OLS program. In Figure 29, the function F_2 is used for both positive and negative flow angles. In Figure 30, the function F_2 is used for negative flow angles and F_1 is used for positive angles. Notice that no calibration values are available for angles greater than 25 degrees or less than -25 degrees. The Processing Program applies the same restriction to A_B so that A_B values greater than -25 degrees are reset to 25 degrees and A_B values less than -25 degrees are reset to -25 degrees.

Once the BLB flow angle is calculated, the velocity magnitude derivation can begin. The algorithm first determines the differential pressure, P(t), which should be measured by a tube pointed directly into the flow. The tube most nearly pointed into the flow is selected and the absolute angle between the flow direction and the tube direction is

$$U(t) = 45 - A_B(t)$$

The conversion function F_P is approximately

$$F_P(U(t)) \cong \frac{q(t)}{P(t)}$$

where q(t) is $q_1(t)$ or $q_2(t)$ as selected and P(t) is the calculated pressure from direct flow into a tube at the same velocity magnitude. Preliminary calibration efforts showed that F_P does not change significantly with velocity magnitude. F_P is a third degree functional approximation of calibration points from all BLB's used in the OLS measurement program.

$$F_P(U(t)) = ((.0000052(U(t) - 20) - .0007158)\,(U(t) - 20) - .0046202)\,(U(t) - 20) + 1.003479$$

As shown in Figure 31, considerable variation is apparent in the calibration points. However, velocity is proportional to the square root of pressure. Thus, a worst case excursion of about 13 percent in pressure ratio becomes about 7 percent excursion in velocity.

Given a pressure measurement from one sensor, q(t), the angle made by the sensor tube with direct flow, U(t), the conversion function, F_P and the air density, as described in Paragraph 6.2.2, $\rho(t)$, the calculation of flow magnitude is straightforward. Since the direct flow pressure is

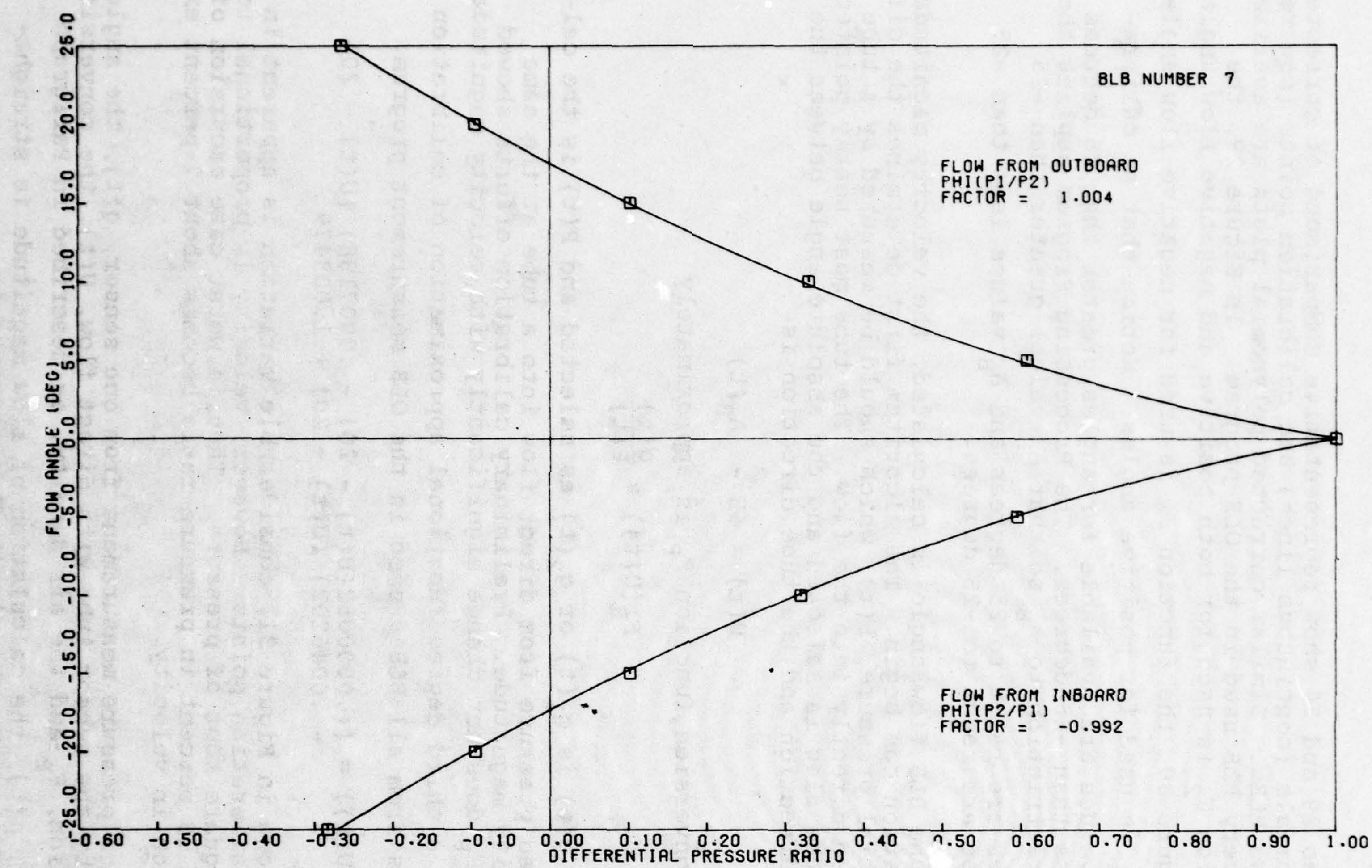


Figure 29. Calibration of BLB number 7.

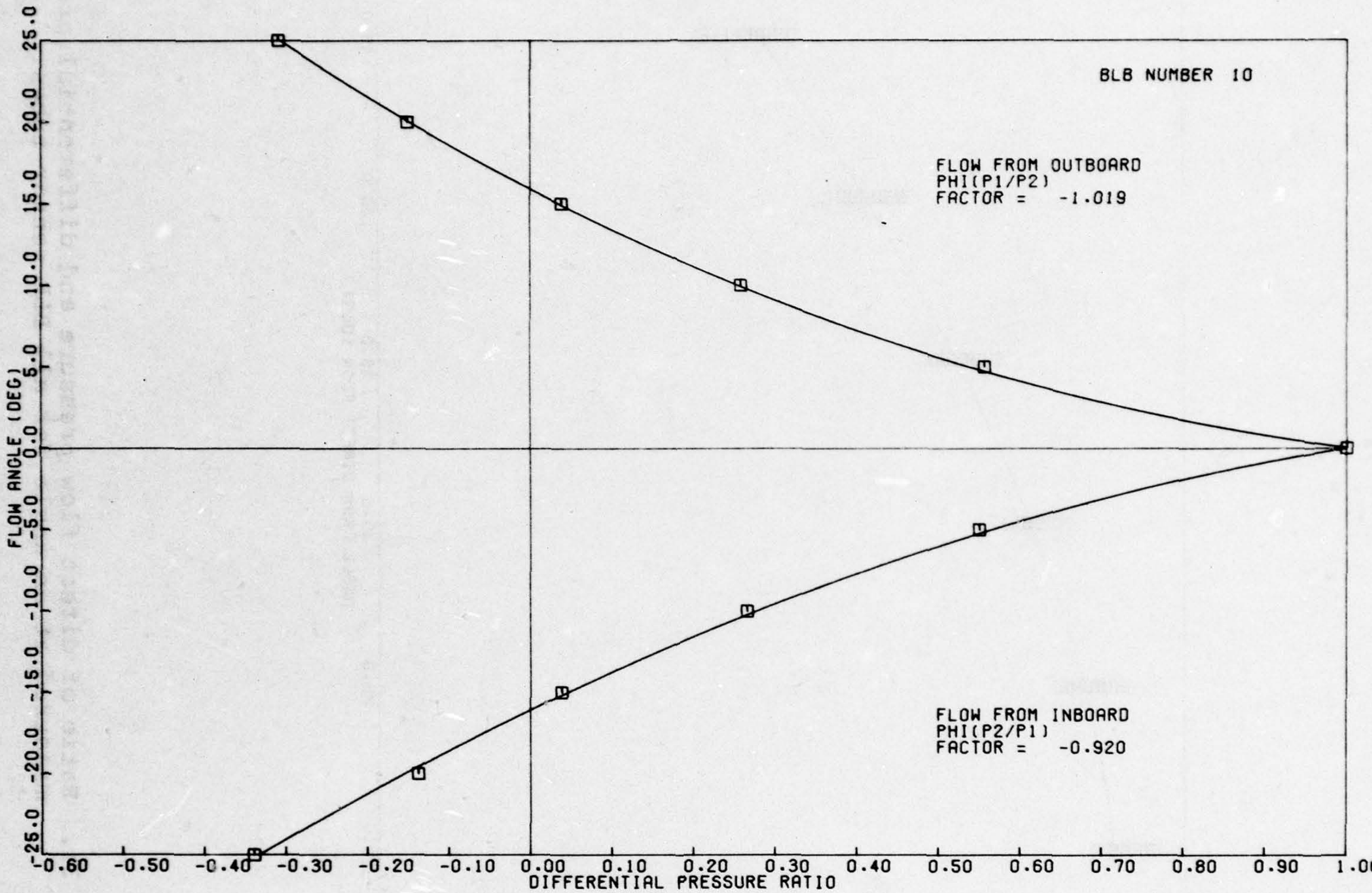


Figure 30. Calibration of BLB number 10.

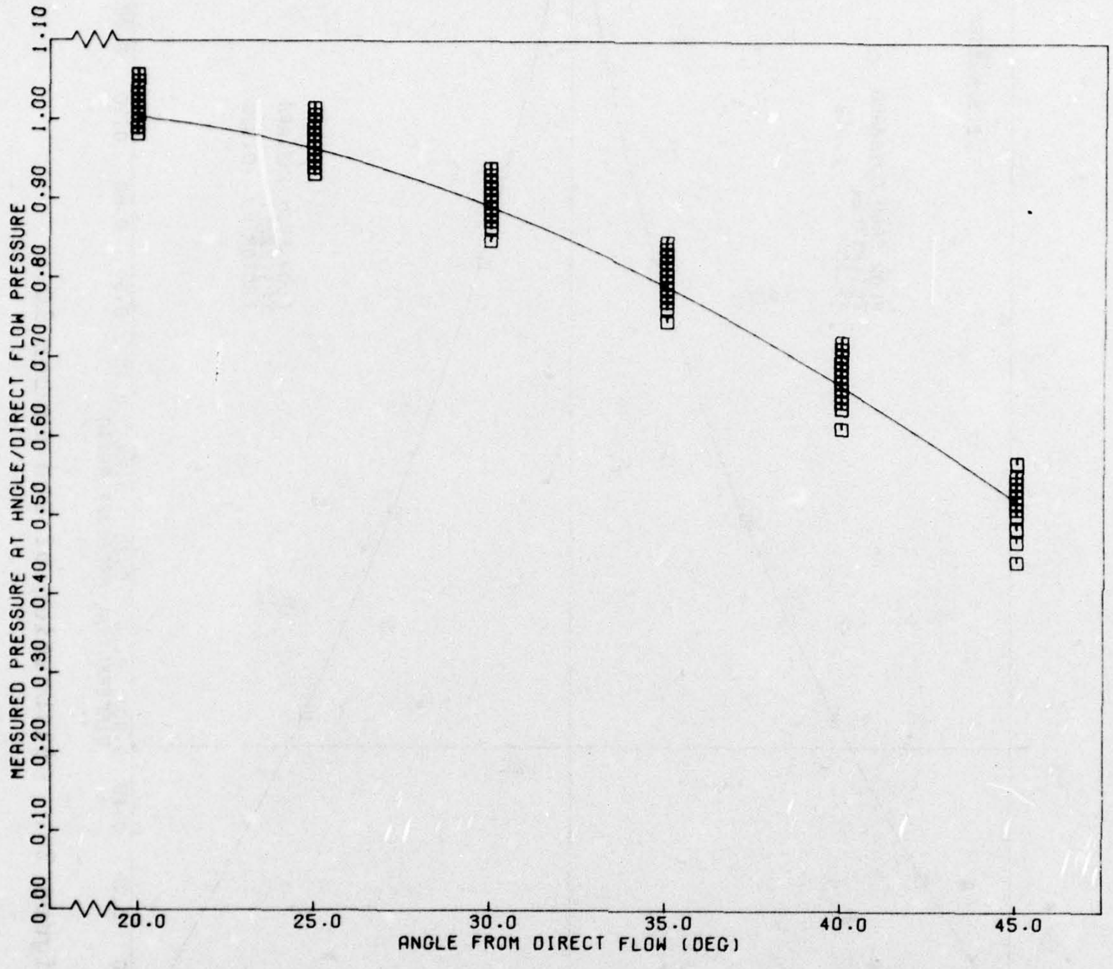


Figure 31. Ratio of direct flow pressure and differential pressure measured at an angle for all BLB sensor tubes.

$$P(t) = 1/2\ \rho(t)\ V^2(t)$$

and

$$P(t) = q(t)/F_P(U(t))$$

then the flow velocity magnitude is

$$V(t) = \sqrt{\frac{2q(t)}{\rho(t)F_P(U(t))}}$$

When $A_B(t)$ exceeds the ±25 degree limitation, the flow is treated as directly into the sensor tube and

$$V(t) = \sqrt{\frac{2q(t)}{\rho(t)}}$$

The BLB flow angle, A_B, is converted to the flow angle relative to the chordline, A_C, by

$$A_C(t) = A_B(t) - (B - 45)$$

The flow magnitude and direction derivations are based on calibrations of the BLBs used in the OLS measurement program. The third-degree polynomial functions F_1, F_2, and F_P are included in the derivation subroutine. The correction factors C_1 and C_2 are stored on the Info File. Processing of BLB data from a different measurement program would require revision of the C_1 and C_2 values stored on the Info File and might require new third degree polynomials to be installed in the derivation subroutine.

6.2.8 Local Blade Displacement

Local blade displacement is computed using measured accelerometer data. Both displacement and slope are computed from harmonic analysis based on main rotor cycles, and a single harmonic at a time may be printed, plotted, or displayed. Letting $\ddot{X}(t)$ be measured acceleration in g's, the oscillatory part of the blade motion is

$$\ddot{X}(t) = \sum_{K=1}^{M} (a_K \cos 2\pi K\omega t + b_K \sin 2\pi K\omega t),$$

$$= \sum_{K=1}^{M} C_K \cos(2\pi K\omega t - \phi_k),$$

where $C_K = \sqrt{a_K{}^2 + b_K{}^2}$, $\phi = \tan^{-1}(\frac{b_K}{a_K})$, and ω is the rotor frequency in hertz. Using the conversion constant C = 386.1 in./(sec^2-g), the displacement for a specific harmonic can be computed as

$$X_K(t) = \frac{CC_K}{(2\pi K\omega)^2} \cos(2\pi K\omega t - \phi_K)$$

in inches. In fact, the harmonic terms C_K and ϕ_K are evaluated using one or more rotor cycles in the same way that the Harmonic Analysis algorithm (Paragraph 6.1.1) can use one or more rotor cycles at the direction of the user. For output, this process creates 256 X(t) values equally spaced in one rotor cycle.

Since the award of this contract, serious deficiencies have been advanced for this method of deriving displacement when the blade is actually rotating. If a rotor speed of 320 RPM and a blade radial station of 260 inches is assumed, the centripetal acceleration at that station is about 760g. Stations nearer the blade root experience proportionately smaller accelerations. If a beamwise accelerometer at a 260-inch station were rotated one degree about a chordwise axis, a measurement of approximately 13g from the 760g centripetal acceleration would be registered. An acceleration amplitude of 13g corresponds to a blade displacement amplitude of about 4.5 inches at 1/rev. Hence, the blade displacement derivation should be used for sensors mounted on a rotating blade with caution.

6.2.9 Local Blade Slope

Local blade slope is interpreted as the derivative of local blade displacements taken along the blade radius for a fixed azimuth. The blade slope algorithm takes as input derived blade displacements for several radial stations along the blade. The slope at each radial station is derived from the quadratic polynomial defined by the displacement at that station and the displacements from the adjacent stations. This process is repeated for every azimuth position represented in the input displacement data. One slope value is generated for every displacement value input.

6.2.10 Density Altitude

Density altitude is computed using the relation

$$H_D = \frac{1-\sigma^{.2349569}}{6.87535(10^{-6})}$$

where σ is air density ratio as described in Paragraph 6.2.2 and the empirical constants are as suggested in Reference 5.

6.2.11 Blade Static Pressure Coefficient

The blade static pressure coefficient, C_p, is computed using the measured parameters blade absolute pressure (psia), boom system static pressure (psia), outside air temperature (°C), and rotor azimuth. In addition, the derived parameters, main rotor RPM, and vehicle true airspeed are required. C_p is computed from

$$C_p = \frac{P_m(t) - P_s(t)}{1/2\rho v^2(t)} K_0 \text{ , where}$$

$P_m(t)$ is blade absolute pressure, $P_s(t)$ is boom system static pressure, ρ is the air density as described in Paragraph 6.2.2, K_0 is the conversion factor 144 in^2/ft^2, and v is the blade station speed given by

$$v(t) = \frac{R_{PM}(t)\ r}{K_1} + K_2 v_T(t) \sin\psi \text{, where}$$

$R_{PM}(t)$ is rotor RPM, r is the blade station radial position, K_1 is conversion from revolution/min to radians/sec (i.e.,

---

[5]Eugene P. Bartlett, First Lieutenant USAF, PERFORMANCE FLIGHT TEST HANDBOOK - PART II, AFFTC-TN-59-22 (ASTIA Document Number AD215865), July 1959.

60/2π), K_2 is conversion from knots to ft/sec (1.688 ft/sec-knot), $v_T(t)$ is vehicle true airspeed and ψ is the rotor azimuth angle.

6.2.12 Blade Normal Force Coefficient

The blade normal force coefficient, C_N, is computed for a fixed radial station and azimuth position from C_p values (Paragraph 6.2.11) for that station. In particular

$$C_N = - \int_o^1 C_p(x, y_u)dx + \int_o^1 C_p(x, y_L)dx$$

where x is the normalized chord station, y_u is the normalized blade surface coordinate in the direction perpendicular to the chordline on the upper blade surface, and y_L is the normalized blade surface coordinate in the direction perpendicular to the chordline on the lower blade surface. For a given chord profile, y_u and y_L are functions of x.

For computational purposes, a finite number of C_p values are available on the upper and lower surfaces of the blade. The algorithm performs the integrations using the trapezoidal method. A C_p value for the x = 0 position is generated by linear extrapolation from the two closest sensors on both the upper and lower surfaces. The two resultant values are then averaged to arrive at a single C_p value for x = 0. A C_p value is also generated for the trailing edge in the same manner using the two sensors closest to the trailing edge for each surface.

6.2.13 Blade Chordwise Force Coefficient

The blade chordwise force coefficient, C_C, is computed for a fixed radial station and azimuth position from C_p values (Paragraph 6.2.11) for that station. In particular,

$$C_C = \int_o^1 C_p(x, y_u)\, y_u'\, dx - \int_o^1 C_p(x, y_L) y_L'\, dx$$

where x,y_u and y_L are described in Paragraph 6.2.12 and y_u and y_L are derivatives with respect to x.

For computational purposes, the integrations are performed using the trapezoidal method. Values of C_p for $x = 0$ and $x = 1$ are estimated using the method described in Paragraph 6.2.12.

6.2.14 Blade Pitching Moment Coefficient

The blade pitching moment coefficient, C_M, is computed with respect to the quarter chord position for a fixed radial station and azimuth position from C_p values (Paragraph 6.2.11) for that station. In particular,

$$C_M = -\int_o^1 (1/4-x)\ C_p(x,y_u)dx + \int_o^1 y_u C_p(x,y_u)y_u'dx$$
$$+ \int_o^1 (1/4-x)C_p(x,y_L)dx - \int_o^1 y_L C_p(x,y_L)Y_L'dx$$

where x, y_u and y_L are described in Paragraph 6.2.12 and y_u' and y_L' are derivatives with respect to x.

For computational purposes, the integrations are performed using the trapezoidal method. Values of C_p for $x = 0$ and $x = 1$ are estimated using the method described in Paragraph 6.2.12.

7. REFERENCES

1. Shockey, Gerald A., Williamson, Joe W., and Cox, Charles R., AH-1G HELICOPTER AERODYNAMIC AND STRUCTURAL LOADS SURVEY, Bell Helicopter Co., USAAMRDL Technical Report 76-39, Eustis Directorate, U.S. Army Air Mobility Research and Development Laboratory, Fort Eustis, Va., February 1977, AD A036910.

2. Ralston, A., and Wilf, H., MATHEMATICAL METHODS FOR DIGITAL COMPUTERS, John Wiley and Sons, New York, 1960, Chapter 24.

3. Eubanks, A. L., FILTER DESIGN AND ANALYSIS WITH APPLICATIONS TO DISCRETE DATA, Bell Helicopter Textron Report 299-099-889, Fort Worth, Texas, 15 August 1977.

4. Yen, J. G., Viswanathan, S., and Matthys, C. G., FLIGHT FLUTTER TESTING OF ROTARY WING AIRCRAFT USING A CONTROL SYSTEM OSCILLATION TECHNIQUE, NASA Symposium on Flutter Testing Techniques, Flight Research Center, Edwards, Calif., October 9-10, 1975.

5. Bartlett, Eugene P., First Lieutenant USAF, PERFORMANCE FLIGHT TEST HANDBOOK - PART II, AFFTC-TN-59-22 (ASTIA Document Number AD215865), July 1959.

APPENDIX A

PROCESSING ERROR NUMBERS

Many user errors are detected as the command step is being entered by the user. Other errors, as well as possible program errors, cannot be detected until processing proceeds. When the program detects an error in processing, an error number is listed and processing of the command step is terminated. For certain errors, a diagnostic message is generated along with the error number. All of these error numbers are cataloged in this appendix. Listed along with each number is the subroutine which detected the error, an indicator '(P)', '(U)' or '(E)', and a short explanation of the error. The indicator (P) specifies a probable program error, (U) indicates a probable user error, and (E) indicates that the source of the error is uncertain.

Currently there are no known processing sequences which would generate an error number with a (P) indicator. However, in a program of this size and complexity, it is impossible to test every conceivable path of execution with every possible data input. In addition, modification of the program to include additional analyses, derivations, output capabilities, or user interface commands could introduce processing errors which would be more easily isolated by those error diagnostics.

NUMBER	ROUTINE	INDICATOR	PROBLEM
1	RMS	(P)	Routine called with an incorrect file number.
2	RMS	(U)	Record wanted outside data area.
3	RMS	(E)	Record wanted outside file area.
4	RMS	(P)	Record size too large.
5	WMS	(P)	Routine called with incorrect file number.
6	WMS	(U)	Record position outside allowed area.
7	WMS	(E)	Record position outside file area.
8	WMS	(P)	Record size too large.
9	FMS	(P)	Routine called with incorrect file number.
10	FMS	(U)	Record wanted outside data area.
11	FMS	(E)	Record wanted outside file area.
101	PROSET	(P)	Bad user interface value specified.
102	PROSET	(P)	Incorrect 'ANALYZE' interface specification.
103	PROSET	(P)	Incorrect 'DERIVE' interface specification.
			May be attempting to read a scratch file which has never had data written into it.
121	COMPSC	(E)	If data have been
122	COMPSC	(E)	written to the scratch
123	COMPSC	(E)	file previously, then
124	COMPSC	(E)	these numbers could indicate a program error.
130	COMPSC	(P)	Incorrect user interface value specified.
135	COMPSC	(U)	A row or column was specified for extraction from a scratch file which does not exist on that file.

NUMBER	ROUTINE	INDICATOR	PROBLEM
151 152 153 154	COMPSC COMPSC COMPSC COMPSC	(P) (P) (P) (P)	Error reading attached parameter data from a scratch file.
160	COMPSC	(U)	Attached parameter data (azimuth, air-speed, RPM, OAT or static pressure) required for a process are not available on the scratch file being read.
165	COMPSC	(U)	A first-dimension scale is specified which conflicts with the available scales.
170	COMPSC	(U)	A first dimension parameter value has been specified which does not occur in the input scratch file.
175	COMPSC	(U)	Type of parameter required for a derivation is not present on the specified scratch file.
180	COMPSC	(U)	A specified first dimension parameter value is beyond the range of stored values.
190	GETDAT	(U)	Specified input data would overflow the available storage area.
195	GETDAT	(U)	The initial data stream required for the process specified is not available.
200	FINDT	(P)	Routine called with illegal parameter specified.

NUMBER	ROUTINE	INDICATOR	PROBLEM
201	FINDT	(U)	A parameter value used to specify input data does not occur for the available data.
202	FINDT	(U)	A parameter used to specify input data is not available.
210	GETI	(U)	Specified input data would overflow the available storage area.
220	GETI	(U)	Insufficient azimuth data to specify number of cycles required for input.
230	INPSET	(P)	Incorrect user interface value specified.
240	INPSET	(U)	Specified counter is not present on the Master File partition.
270	INFSCR	(U)	No data present in scratch file specified for input.
280	RTRVSC	(E)	Data to be read from scratch file would overflow program storage.
304	AMPSET	(U)	An insufficient number of points were provided to generate an amplitude spectrum.
325	TSAV1	(P)	Improper file specified for random access write.
326 327	TSAV1 TSAV1	(U) (U)	Processed data overflows temporary scratch file area.

NUMBER	ROUTINE	INDICATOR	PROBLEM
328	TSAV1	(P)	Improper record size specified for random access write to temporary scratch file.
345	TSAV1	(P)	Improper file specified for random access write to temporary scratch file.
346	TSAV1	(U)	Processed data over flows SCF1 or SCF2.
347	TSAV1	(U)	
348	TSAV1	(P)	Improper record size specified for random access write to SCF1 or SCF2.
350	PRO1	(P)	Bad user interface value specified.
365	TSAV2	(P)	Improper file specified for random access write.
366	TSAV2	(U)	Processed data overflows SCF1 or SCF2.
367	TSAV2	(U)	
368	TSAV2	(P)	Improper record size specified for random access write to SCF1 or SCF2.
375	TSAV2	(P)	Improper file specified for random access write to temporary scratch.
376	TSAV2	(U)	Processed data over flows SCF1 or SCF2.
377	TSAV2	(U)	
378	TSAV2	(P)	Improper record size specified for random access write to temporary scratch file.
380	TSAV2	(P)	Incorrect process output sequence.

NUMBER	ROUTINE	INDICATOR	PROBLEM
390	TSAV2	(U)	Column scale specified for output with single column element available.
401	MULTPL	(P)	Incorrect file specified in reading data from direct access temporary scratch.
402	MULTPL	(P)	Incorrect record number.
403	MULTPL	(P)	Specified for read from direct access temporary scratch file.
404	MULTPL	(P)	Incorrect record size specified for read from direct access temporary scratch file.
430	SINGPL	(U)	Attached parameter values are missing for generation of output scale.
440	DISPOS	(U)	APLOT specified when a plot scale generated by MPLOT is not already present.
460	OUTSET	(P)	Incorrect user interface value specified.
470	OUTSET	(U)	Plot output is specified when the process output will consist of a single point.
475	OUTSET	(U)	The output mode selected provides for a different number of independent variable dimensions than the output data will possess.

NUMBER	ROUTINE	INDICATOR	PROBLEM
480	ALLATT	(E)	Error reading measured data to generate OAT, true airspeed or static pressure.
490	AZIMTH	(U)	Insufficient azimuth data are available to define a stream specified in cycles.
500	IMPATT	(P)	Incorrect value in Info record for data on the Master File.
510	ATTGET	(E)	Error generating azimuth times.
520	ATTGEN	(U)	Insufficient airspeed, OAT, or static pressure data for requested display.
530	ATTGEN	(U)	Data generated overflows available program storage.
540	AZMGEN	(U)	Insufficient azimuth data for requested display.
550	AZMGEN	(U)	Data generated overflows available program storage.
560	POWGEN	(U)	Insufficient azimuth, airspeed, OAT, or static pressure data for the requested derivation.
570	POWGEN	(P)	Error interpolating data.
580	SCALGN	(U)	Insufficient azimuth or airspeed for requested scale.

NUMBER	ROUTINE	INDICATOR	PROBLEM
600	COMPGP	(U)	Invalid Info file group name specified.
610	COMPGP	(U)	Specified Info file group name not specified.
620	COMPGP	(U)	Info file group is incomplete.
630	COMPGP	(U)	Info file group has line error.
640	COMPGP	(U)	Info file group has syntax error.
650	COMPGP	(P)	Incorrect specification from user interface.
680	DFILTR	(U)	Too few data points provided for filtering process.
690	HARMNY	(U)	The input data stream contained two double-row elements. The Harmonic Analysis process can handle only one double-row element.
700	HARMNY	(U)	The input data stream does not span a full rotor cycle.
710	HARMNY	(U)	The input data sample rate is too low to calculate the specified harmonics.
740	DSPSET	(U)	A BOTTOM double-row element was specified for a blade displacement derivation.

NUMBER	ROUTINE	INDICATOR	PROBLEM
752	DSPSET	(U)	Insufficient accelerometer data to generate blade displacement for one complete rotor cycle.
753	DSPSET	(U)	Accelerometer data sample rate is too low to calculate blade displacement for the specified harmonic.
771	DAMPST	(U)	Input data stream for moving block damping is too short or has too low a sample rate.
780	PRCFST	(U)	Total data requirements overflow available program storage.
785	PRCFST	(U)	Insufficient attached parameter data for Cp derivation.
790	FLWSET	(U)	Input data stream overflows available program storage.
795	FLWSET	(U)	Insufficient attached parameter data for Blade Local Flow Magnitude or Direction derivation.
800	YSFRST	(E)	Generated contour or surface plot matrix will overflow available storage.
805	YSFRST	(U)	Intermediate data matrix will overflow available storage in generating contour or surface plot.

NUMBER	ROUTINE	INDICATOR	PROBLEM
811	YSFRST	(P)	Temporary scratch file read error while generating a data matrix for a surface or contour plot.
812	YSFRST	(P)	
813	YSFRST	(P)	
814	YSFRST	(P)	
825	TSAV3	(E)	SCF1 or SCF2 write error. Possible over flow.
826	TSAV3	(E)	
827	TSAV3	(E)	
828	TSAV3	(E)	
831	TSAV3	(P)	Temporary scratch file read error.
832	TSAV3	(P)	
833	TSAV3	(P)	
834	TSAV3	(P)	
835	TSAV3	(P)	Temporary scratch file write error.
836	TSAV3	(P)	
837	TSAV3	(P)	
838	TSAV3	(P)	
850	CONSET	(U)	Cylindrical format 3-D plot requested with first independent variable other than azimuth.
860	CONREC	(U)	3-D plot requested in nongraphics mode.
871	GETEMP	(P)	Temporary scratch file read error.
872	GETEMP	(P)	
873	GETEMP	(P)	
874	GETEMP	(P)	
880	GETEMP	(U)	Input data overflows available program storage.
921	CYAVST	(P)	Program errors in generating cycle averaged data.
922	CYAVST	(P)	
923	CYAVST	(P)	
924	CYAVST	(P)	
940	CYAVST	(U)	Complete cycle of data not available.
950	MINXST	(P)	Cycle start time does not match start of data.

NUMBER	ROUTINE	INDICATOR	PROBLEMS
960	MINXST	(U)	Complete cycle of data not available.
970	SLOPST	(U)	Data overflows available program storage.